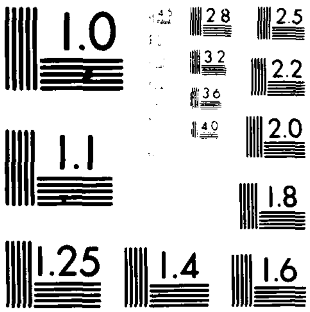
AD-A108 107 OHIO STATE UNIV RESEARCH FOUNDATION COLUMBUS F/G 1/3
DEVELOPMENT OF A COMPUTER BASED PRESENTATION OF NON-STEADY HELI--ETC(U)
SEP 81 H R VELKOFF, R R NAVARRO DAAG29-79-C-0074
UNCLASSIFIED ARO-14142.6-EX NL

1 OF 2
AD
A108107



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963 A

RF Project 761646/711999
Interim Report

the
ohio
state
university
research foundation
1314 kinnear road
columbus, ohio
43212

DEVELOPMENT OF A COMPUTER BASED PRESENTATION OF NON-STEADY HELICOPTER ROTOR FLOWS

Henry R. Velkoff and Richard R. Navarro
Department of Mechanical Engineering

U.S. ARMY RESEARCH OFFICE
P.O. Box 12211
Research Triangle Park, North Carolina 27709

Contract No. DAAG29-79-C-0074

September, 1981

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A108 107

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
DEVELOPMENT OF A COMPUTER BASED PRESENTATION OF NON-STEADY HELICOPTER ROTOR FLOWS

5. TYPE OF REPORT & PERIOD COVERED
Interim
July 1, 1979 - April 30,1981

6. PERFORMING ORG. REPORT NUMBER
761646/711999

7.
Henry R. Velkoff and Richard R. Navarro

8. CONTRACT OR GRANT NUMBER(s)
Contract DAAG29-79-C-0074

9. PERFORMING ORGANIZATION NAME AND ADDRESS
The Ohio State University
Research Foundation, 1314 Kinnear Road
Columbus, Ohio 43212

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
P-14142-E

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709

12. REPORT DATE
September, 1981

13. NUMBER OF PAGES
118

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*
Approved for public release; distribution unlimited.

17. DISTRIBUTION ST. ENT *(of abstract entered in Block 20, if different from Report)*
NA

18. SUPPLEMENTARY TES
The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*
Helicopter Rotor
Rotor Flows
Rotor Wake
Hot-wire Anemometry

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A system tc measure the non-steady velocities in the model rotor wake was modified to provide velocity data in a form more useful to the user of the data. Hot wire angle parameters were modified so as to provide better correlations by using cosine law in place of the usual K^2 technique common in hot-wire anemometry. The results of the non-steady measurements were presented in the form of mction pictures of the non-steady computer generated vector plots.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

Interim Report

Development of a Computer Based Presentation of Non-Steady Helicopter Rotor Flows

September 1981

Henry R. Velkoff

Richard R. Navarro

Department of Mechanical Engineering
The Ohio State University
Columbus, Ohio

Abstract

A system to measure the non-steady velocities in the model rotor wake was modified to provide velocity data in a form more useful to the user of the data. Hot wire angle parameters were modified so as to provide better correlations by using cosine law in place of the usual K^2 technique common in hot-wire anemometry. The results of the non-steady measurements were presented in the form of motion pictures of the non-steady computer generated vector plots.

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to Professor H. R. Velkoff for his undaunted moral support and invaluable technical contributions without which the work herein would not have been the success that it was. Appreciation is also expressed for the financial sponsorship of the U.S. Army Research Office, Durham, North Carolina.

TABLE OF CONTENTS

LIST OF FIGURES

Figure page

LIST OF SYMBOLS

A	3-D hot-film sensor
A/D	analog to digital converter
B	3-D hot film sensor
b	number of blades
C	3-D hot-film sensor
c	blade chord
C_0, C_1, C_2, C_3, C_4	calibration polynomial coefficients
E	anemometer output voltage
K	directional sensitivity coefficient
P	directional sensitivity exponent
R	rotor radius
R_{LW}	lead wire resistance
R_{OP}	operating resistance of sensor
R_{POT}	resistance of potentiometer in bridge
U	flow vector
U_A, U_b, U_c	components of flow vector in direction of A, B, C
V_A, V_B, V_C	the effective cooling velocity for sensors A, B, C
V_{eff}	effective cooling velocity
V_x, V_y, V_z	components of flow vector, U, in tunnel directions x,y,z
X	axial tunnel coordinate direction
Y	vertical tunnel coordinate direction
Z	lateral tunnel coordinate
α	angle between flow vector and sensor A
β	angle between flow vector and sensor B

γ	angle between flow vector and sensor C
μ	advance ratio
ϕ_A,ϕ_B,ϕ_C	angle between flow vector and normal to A, B, C
ψ	blade angular position

CHAPTER I

INTRODUCTION

Helicopter rotor wake characteristics during low speed terrain flight have become the subject of increased interest in recent years, particularly in military applications.

The interaction of blade tip vortices at transition speeds, ($0 < \mu < 0.2$) and the resulting wake patterns are significant in terms of their effect on aerodynamic surfaces such as wings, stabilizers, or fuselage. The apparent inability of theoretical methods to accurately predict wake velocities in the transition flight regime has led to many experimental investigations, Refs. (5,6,8).

This work is an extension of the experimental investigation of Ref. (8), in which the measurement capability of non-steady rotor wake velocities was established. Presented here are the computer based non-steady rotor wake velocity data reduction and presentation techniques.

The massive amounts of data being manipulated necessitated the use of a computer facility. The system used here consists of a DEC PDP 11/60 minicomputer and a GENRAD analog to digital converter. Flow measurements were made in a 4 x 8 ft. wind tunnel, of a 2-bladed model helicopter rotor using a 3-dimensional hot film anemometer system.

CHAPTER II

EXPERIMENTAL APPARATUS

2.1 The Test Rig

The data for this work was acquired using the Mechanical Engineering Department 4 ft. x 8 ft. wind tunnel facility at The Ohio State University. The test section contained a 2.5 ft. dia., two-bladed, teetering rotor. Collective pitch and rotor shaft tilt angle were adjustable but once set for a test, they remain constant. The rotor details are listed in Table 2.1.

Table 2.1. Details of Rotor

Number of Blades	2
Rotor Radius, ft.	1.25
Blade Chord, in.	2.1875
Rotor Solidity bc/πR	.0928
Root Cutout %R	12.08
Blade Taper Ratio	1
Blade Twist	8° (from root to tip)
Airfoil Section	NACA 0012

Rotor speed is variable up to a maximum of 3000 RPM yielding a maximum tip speed of 400 ft/sec. The wind tunnel is capable of stream velocities up to a maximum of 75 ft/sec. Figure 2-1 shows the test section and the rotor orientation within the test section.

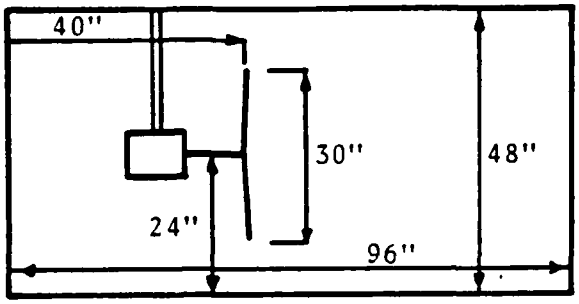


Figure 2-1. Rotor Location in Test Section Looking Downstream

2.2 The Hot-Film Anemometry

The rotor induced velocities were measured by means of a constant temperature, 3-dimensional hot-film anemometer system incorporating a TSI model 1294-60-18 probe, and 3 DISA 55005/102C control circuits. The DC power supply for the control circuits and signal amplifiers is a SOLA ± 15V, 1.6A model 83-15-2216.

This anemometry system used in these tests differs from that of Ref. (8) in that changes became necessary in order to eliminate severe calibration drift problems encountered at the onset of this work. The system as used here is shown in Figure 2-2. The differences between this and the system of Ref. (8) include: 1) the elimination of lead compensators; 2) the use of .006 in. diameter probe as opposed to .002 in. diameter; 3) the substitution of 3 two stage integrated circuit signal amplifiers for the 3 DANA 2210 amplifiers and; 4) minor modification of the DISA control circuits. The lead compensators were removed for two reasons. First, it was suspected that they may have been responsible for the drift phenomenon. Though they may have

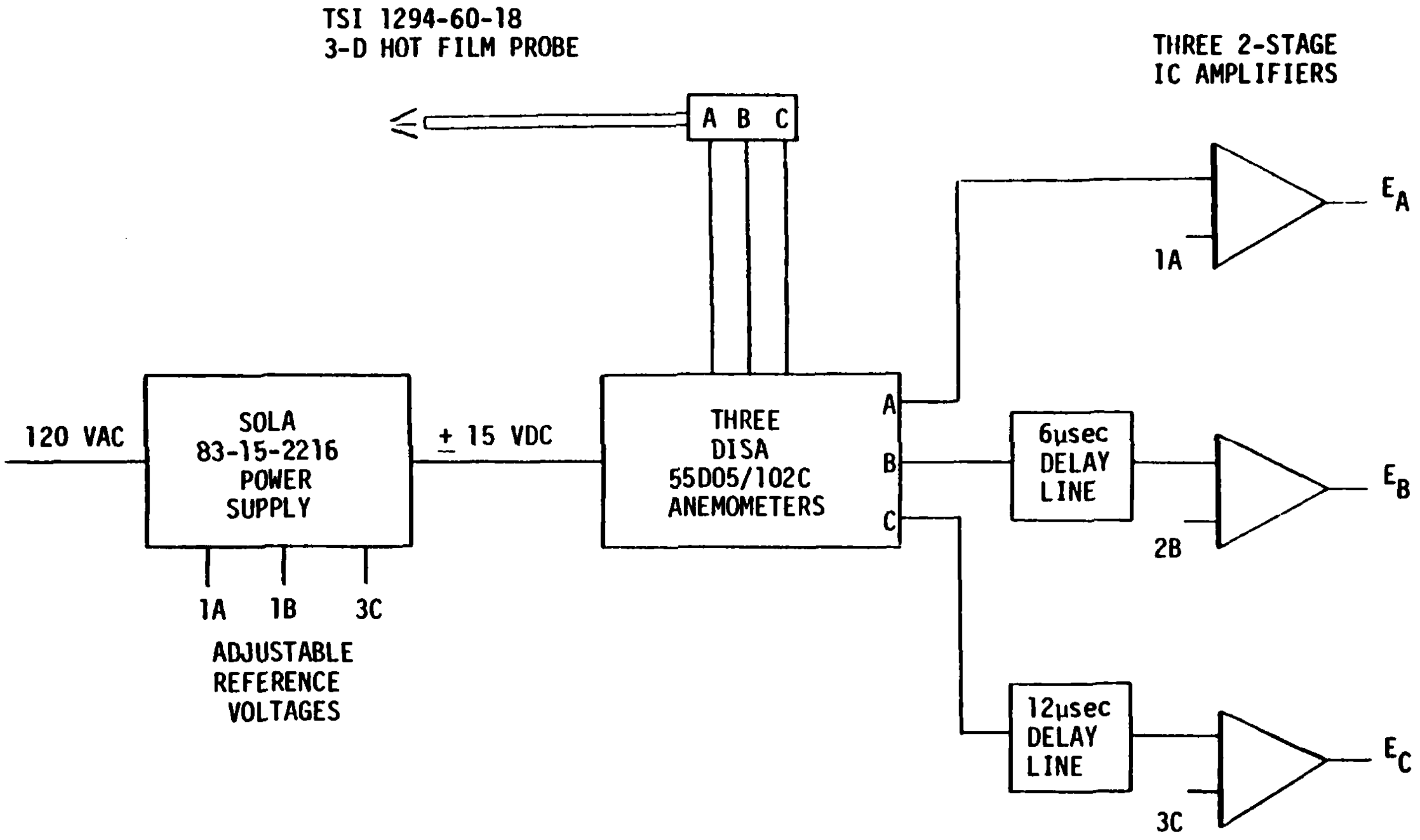


Figure 2-2. Block Diagram of the Anemometer System

contributed to this problem, their removal did not solve the problem. Secondly, their effectiveness in extending frequency response was being negated by frequency limiting software techniques which were employed in an effort to reproduce results previously obtained. Details of this software approach are included in Appendix A.

The three integrated circuit amplifiers provided the variability needed in controlling the output voltage range for the range of input velocities, 0 to 65 ft/sec. The range of amplification is continuously variable from a factor of 1.0 to 100. The DANA units have discreet amplification factors making it necessary to add voltage dividers for "fine tuning". This is particularly cumbersome since changing probes necessitates a change in the voltage divider. Figure 2-3 shows the new amplifier circuit. Note here also that a bias voltage adjustment potentiometer is provided.

After an attempt at designing a completely new anemometer control circuit met with only moderate success, see Appendix B, the existing DISA control circuitry was repaired. All transistors in Channel A were replaced, see Figure 2-4, as was the TD101 transistor pair in Channel C. It was also discovered that the capacitor between the base of the 2N3906 transistor and ground has a significant effect on the frequency response and output drift of the system. After trying several values in the range between .056μF and .22μF a value of .15μF was chosen. The other modification was the addition of a 100Ω OHMITE model 4092 resistor in series with the 1K potentiometer in the bridge. This was necessary to accommodate a probe which had an operating resistance, R_{op}, equal to 10.13Ω. Since the overheat ratio is obtained

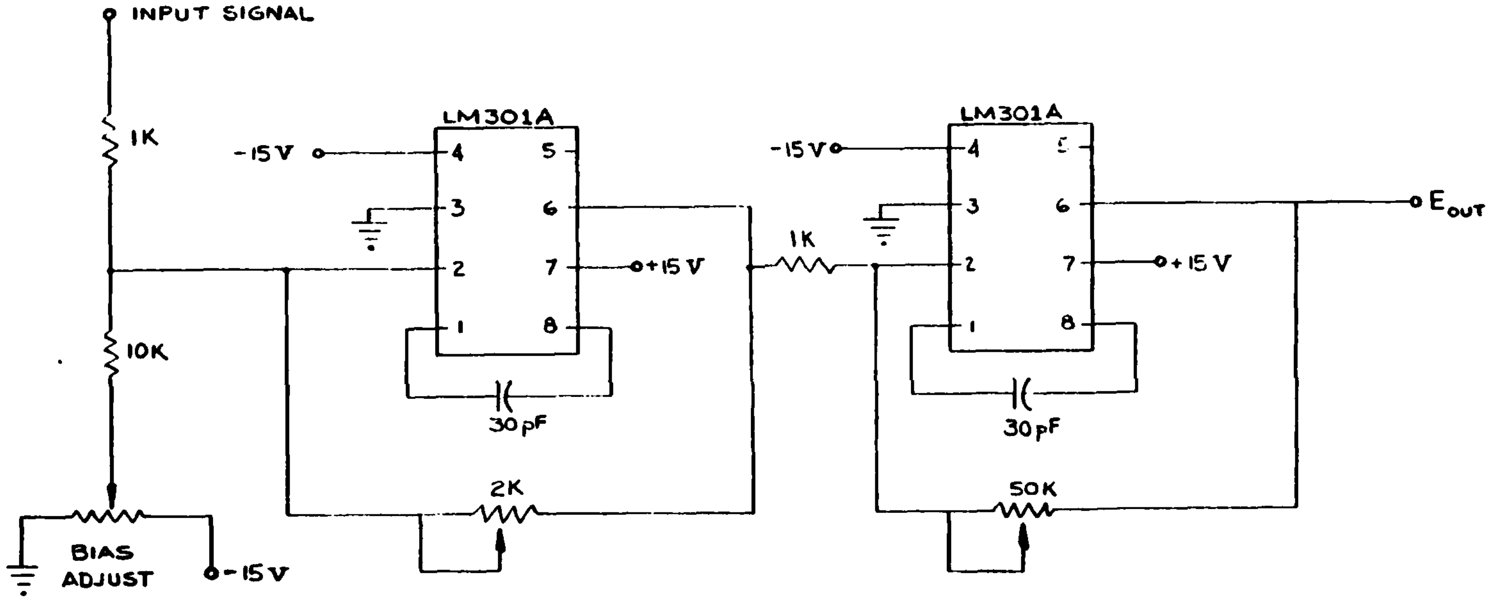


Figure 2-3. Two Stage IC Amplifier Circuit

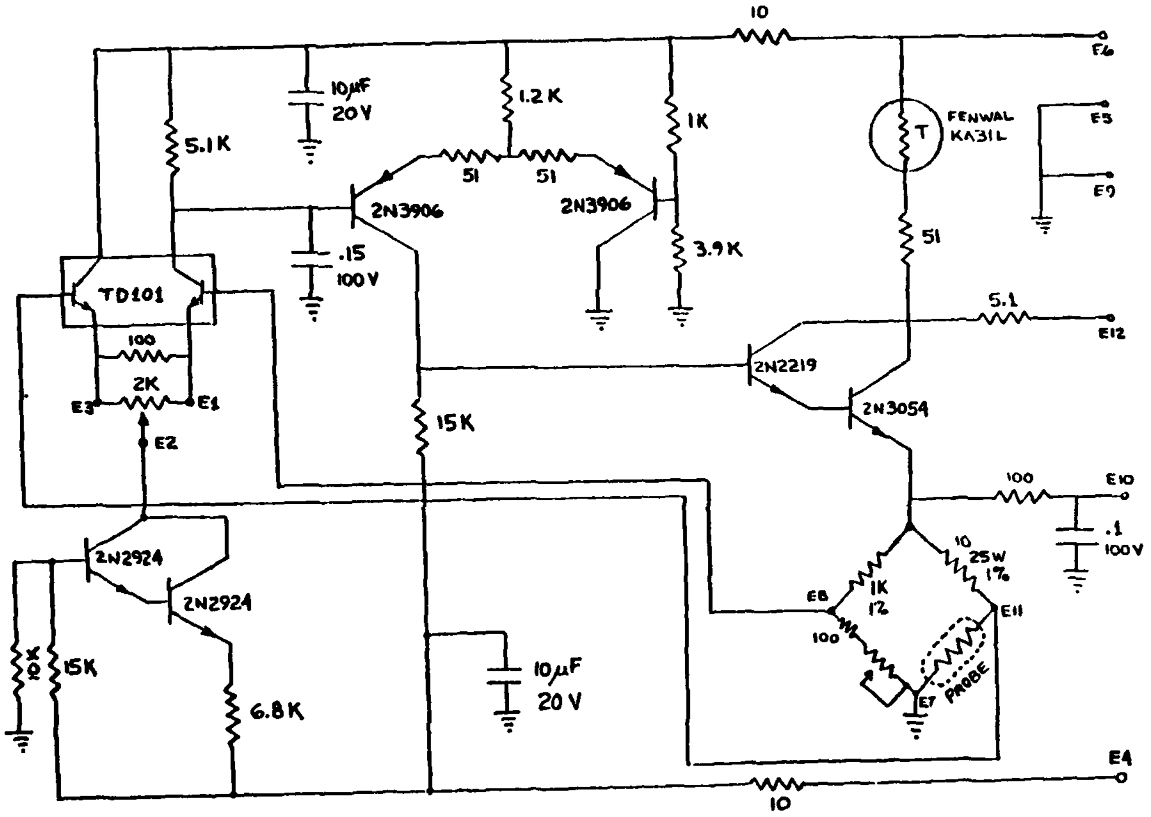


Figure 2-4. DISA 55D05/102C Anemometer Control Circuit

by multiplying the sum of R_{OP} and $R_{lead\ wire}$ by 100, the resistance in this leg of the bridge must be

$$R_{POT} = (R_{OP} + R_{LW}) \times 100$$

$$= (10.13 + .30) \times 100$$

$$R_{POT} = 1043\Omega$$

With the addition of the 100Ω series resistor this value of R_{POT} can be reached.

The anemometer circuitry as described here should be considered only marginally adequate for the task being performed. A new circuit of the same design would improve the reliability and stability.

2.3 Data Acquisition

2.3.1 The Probe Traverse Mechanism

The probe is mounted on a traverse mechanism which allows the operator to position the probe at any point in the test volume. The spatial orientation of the probe in the tunnel is depicted in Figure 2-5. The traverse mechanism employs motorized positioning in the Y and Z-directions and a manually operated pully arrangement for the X-direction. The test volume measures 68 in. x 40 in. x 15 in., (X, Y, and Z respectively). In the X-direction data is taken at each 4 in. increment starting at X = 28 in. going to X = - 40 in. In the Y-direction data is taken at 4 in. increments starting at Y = 20 in. going to Y = - 20 in. In the Z-direction data is taken at 3 in. increments starting at Z = 12 in. going to Z = - 3 in. Thus the test volume for a complete data set contains a matrix of 1188 points.

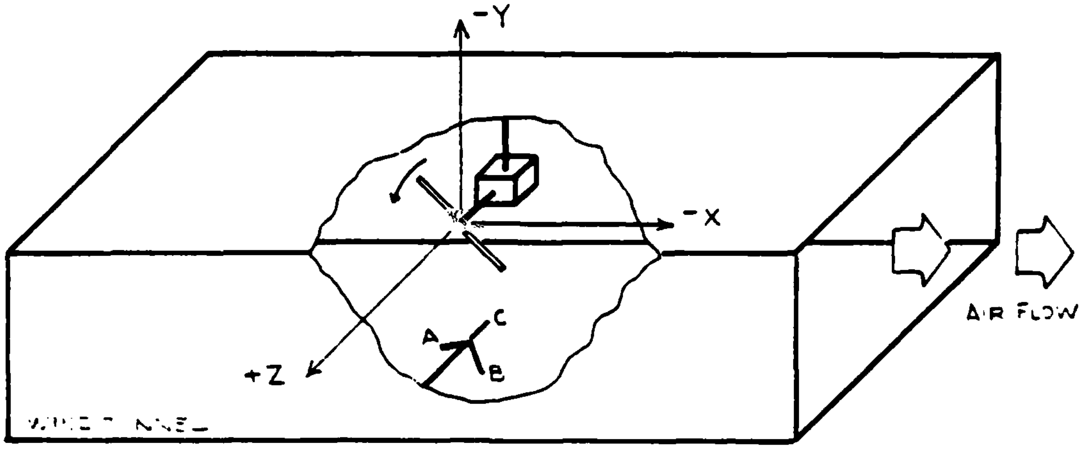


Figure 2-5. The Spatial Orientation of the Probe in the Wind Tunnel

However, due to rotor interference, 147 points cannot be measured as indicated by the shaded region in Figure 2-6. Note also that the rotor hub center is the point <0,0,0> for a shaft tilt angle equal to zero.

2.3.2 The Computer Interface

The anemometer system is linked to the DEC PDP 11/60 minicomputer via an interface circuit and a GENRAD A/D converter. The interconnection of the two systems is shown in Figure 2-7. The inclusion of the delay lines in Channels B and C is to allow for the read time cycle of the A/D. Without the delay lines, A, B, and C would be read at slightly different points in the rotor revolution.

The interface circuit is actually a pulse shaping circuit which triggers the A/D. The interface receives pulses from two magnetic pickups mounted on the rotor support structure. One pickup senses each tooth of a 72 tooth gear turning with the rotor. The second pickup is

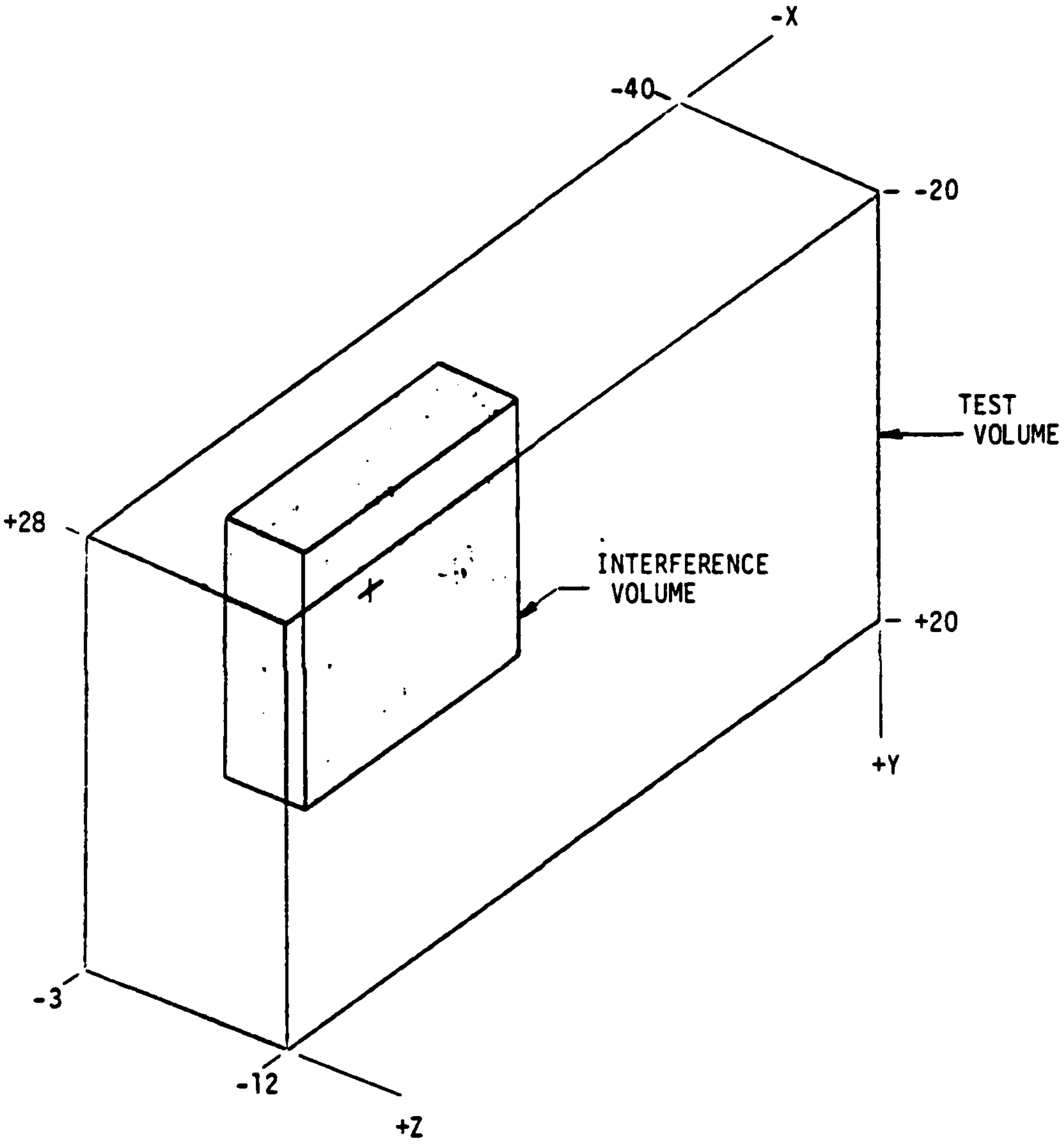


Figure 2-6. Rotor Interference Volume Relative to Test Volume

a once per revolution signal. The output of the interface is a pulse train which triggers the A/D to sample sensors A, B, and C 72 times per revolution (every 5 degrees) for as many revolutions as required. In this case 14 revolutions were used giving a total of 72 x 14 x 3 or 3024 pieces of data at each point in the test volume. Assuming then that the signal is periodic, an average is obtained so that only 72 x 3 or 216 pieces of data are actually stored in computer memory. A further reduction in the amount of memory space needed is accomplished through a double precision integer coding scheme. Essentially this technique codes the velocity V_x in the first 3 digits of a 9-digit integer, V_y in the second 3 and V_z in the last 3. This uses less memory space than 3 separate real numbers.

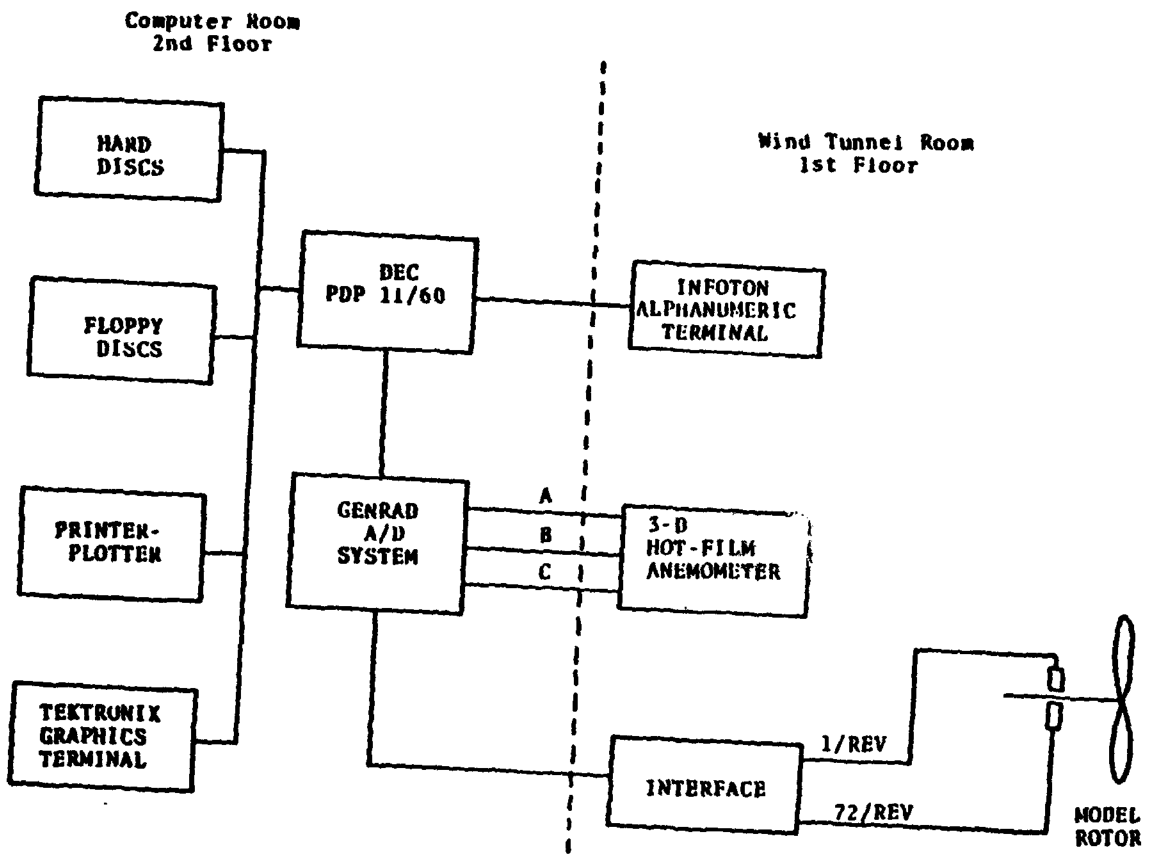


Figure 2-7. Block Diagram of the Entire Measuring System

CHAPTER III

PROBE DIRECTIONAL SENSITIVITY AND CALIBRATION

3.1 Directional Sensitivity of Cylindrical Sensors

The output of the hot-film anemometer is a voltage which is related to the fluid velocity as:

$$E^2 = A + BU^{1/2} \tag{3-1}$$

where U is the stream velocity, E is the output voltage, and A and B are constants determined by the circuitry and the physical properties of the sensor and fluid medium. However, if the fluid velocity is not perpendicular to the sensor, equation 3-1 does not hold. To overcome this problem, an "effective cooling velocity" can be defined as the perpendicular velocity which produces the same cooling effect as the actual, non-perpendicular velocity. This effective cooling velocity must then be correlated to the true stream velocity. The relationship used in Ref. (6) is given as:

$$V_{eff}^2 = U^2(\cos^2\phi + K^2\sin^2\phi) \tag{3-2}$$

where:

V_{eff} = effective cooling velocity

U = stream velocity

ϕ = angle between the free stream vector and the normal to the sensor. See Figure 3-1.

K = dimensionless constant

The term $K^2\sin^2\phi$ accounts for the cooling effect of flow along the sensor.

Another correlation which is used is that of Ref.   given by:

$$V_{eff} = U \cos^P \phi \qquad (3\text{-}3)$$

where the exponent, P, is used to account for cooling effects of flow along the sensor. This relationship is used here in order to simplify the data reduction procedure.

3.2 Method of Data Reduction

The aim of the data reduction procedure is to derive from the voltage outputs of each sensor the spatial orientation of the stream vector in the tunnel. It is therefore critically important that the angles, α, β, and γ of Figure 3-2 be accurately determined.

Using the method of 3.1, each of the 3 mutually perpendicular sensors will have an equation for the effective cooling velocity across it.

$$V_{effA} = V_A = U \cos^P \phi_A \qquad (3\text{-}4)$$

$$V_{effB} = V_B = U \cos^P \phi_B \qquad (3\text{-}5)$$

$$V_{effC} = V_C = U \cos^P \phi_C \qquad (3\text{-}6)$$

Since V_A, V_B, V_C, and P (note that p may be different for each sensor, but are used as the same value) will be known from the calibration, U must now be found in terms of these quantities.

In order to make use of the geometric constraint that:

$$\cos^2\phi_A + \cos^2\phi_B + \cos^2\phi_C = 2 \qquad (3\text{-}7)$$

equations 3-4 to 3-6 are arranged in the form

$$\left(\frac{V_i}{U}\right)^{1/P} = \cos\phi_i \qquad i = A,B,C \qquad (3\text{-}8)$$

It then follows that

$$\Sigma\left[\left(\frac{V_i}{U}\right)^{1/P}\right]^2 = 2 \qquad (3\text{-}9)$$

and in expanded form this becomes

$$\frac{1}{U^{2/P}}\left[V_A^{2/P} + V_B^{2/P} + V_C^{2/P}\right] = 2 \qquad (3\text{-}10)$$

which can be solved for U

$$U = \left[\frac{V_A^{2/P} + V_B^{2/P} + V_C^{2/P}}{2}\right]^{P/2} \qquad (3\text{-}11)$$

Now in order to find the angles ϕ_A, ϕ_B, and ϕ_C which will locate U in the probe system of coordinates, equations 3-4 to 3-6 will be solved for these angles.

$$\phi_A = \cos^{-1}\left(\frac{V_A}{U}\right)^{1/P} \qquad (3\text{-}12)$$

$$\phi_B = \cos^{-1}\left(\frac{V_B}{U}\right)^{1/P} \qquad (3\text{-}13)$$

$$\phi_C = \cos^{-1}\left(\frac{V_C}{U}\right)^{1/P} \qquad (3\text{-}14)$$

With ϕ_A, ϕ_B and ϕ_C known, the vector U has been located in probe coordinate system.

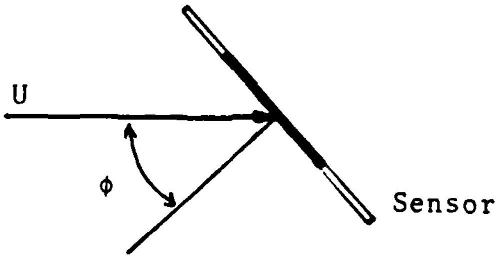


Figure 3-1. Inclined Cylindrical Sensor

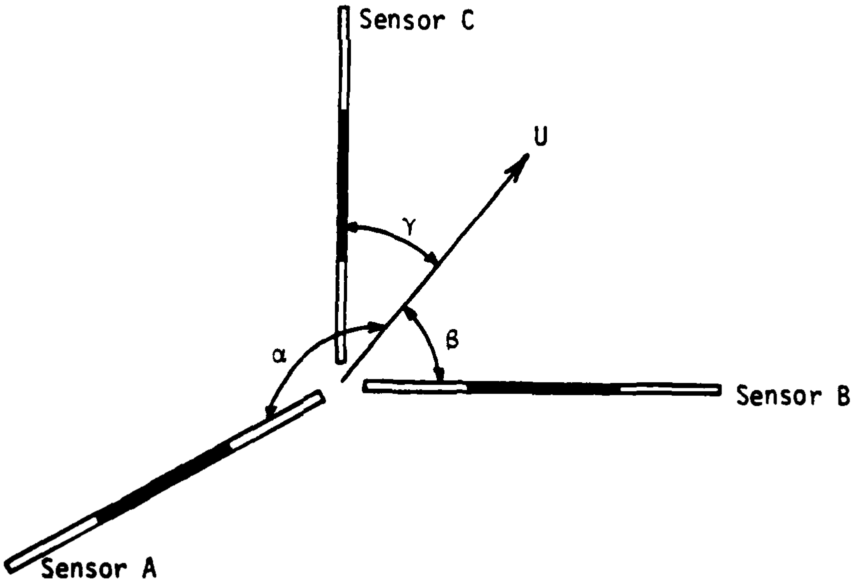


Figure 3-2. Geometry of 3-Dimensional Hot Film Probe Tip

3.3 Transformation of Velocities from the Probe to the Tunnel Coordinate System

Due to the nature of the probe geometry and its orientation within the tunnel, the vector U must be transformed into its X, Y, and Z components in the tunnel coordinate system shown in Figure 3-3. Though the probe axis is parallel to the tunnel Z direction the individual sensors are each inclined at an angle of 54.74° to the probe axis. Figure 3-3 also shows the vectors OA, OB, and OC which represent flow in the direction of sensors A, B, and C respectively. Note also that the vector OC lies in the X-Z plane.

If U_A, U_B, and U_C are defined as the components of U in the direction of sensors A, B, and C respectively their equations are:

$$U_A = U\sin\phi_A \tag{3-15}$$

$$U_B = U\sin\phi_B \tag{3-16}$$

$$U_C = U\sin\phi_C \tag{3-17}$$

where it should be recalled that ϕ_i is defined as the angle between U and the normal to the sensor.

From the geometry of the probe and its location with respect to X, Y, and Z in Fig. 3-3 the angles EOC, EOD, AOD and BOD are:

$$<EOC = 35.26°$$

$$<EOD = 54.74°$$

$$<AOD = 45°$$

$$<BOD = 45°$$

Using these angles the transformation equations 3-18 through 3-20 are defined as

$$V_x = - 0.40825(U_A + U_B) - 0.8165\ U_C \qquad (3\text{-}18)$$

$$V_y = 0.7071(U_A - U_B) \qquad (3\text{-}19)$$

$$V_z = 0.57735(U_A + U_B - U_C) \qquad (3\text{-}20)$$

The quantities V_x, V_y, and V_z are the components of the vector U in the tunnel coordinate directions x, y, and z.

3.4 Probe Calibration

3.4.1 Velocity Calibration

The velocity calibration of the probe was performed using a calibration air chamber with an exit nozzle diameter of 0.75 in. The chamber static pressure is measured using an inclined manometer from which the nozzle exit air velocity is determined. Each sensor is in turn positioned such that its support prongs lie in the horizontal plane and the sensor is within 0.5 in. of the vertical exit plane of the nozzle and parallel to it. The jet velocity is subsequently varied from 0 to 65 ft/sec. as the output voltage is recorded for each velocity. A total of 13 velocities are used in the range from 0 to 65 ft/sec and a canned computer program is used to fit a fourth order polynomial through these points. The velocity calibration curve for channel A used in this work is shown as an example in Fig. 3-4. The equations for this curve and similar curves for B and C are entered in the computer program, COSPX.FTN (Appendix C), and are of the form

$$V_{eff} = C_4E^4 + C_3E^3 + C_2E^2 + C_1E + C_0 \qquad (3\text{-}21)$$

3.4.2 Directional Sensitivity Calibration

In previous work the directional sensitivity of the probe sensors has been accounted for by use of the constant, K^2, in equation 3-2. This work employs the constant, p, of equation 3-3 as a result of calibration data which suggested that this method would work well.

The angular calibration data taken is independent of which method is used. The probe is positioned on the calibrator in the same manner as in the velocity calibration, only now the angle ϕ (see Fig. 3-1) is varied from 0° to 65° in 5° increments. The angle ϕ was not varied beyond 65° due to suspected interference of the sensor support prongs. The output voltage at each angle is recorded and this process is repeated for 8 velocities in the range from 10 ft/sec to 65 ft/sec.

At this stage a computer program is used to calculate V_{eff}/U for each angle ϕ using equation 3-21. It was decided to fit the equation,

$$\frac{V_{eff}}{U} = \cos^{p}\phi \tag{3-22}$$

to the V_{eff}/U data at the $\phi = 65°$ point by choosing an appropriate value for p. The calibration data indicated that $\frac{V_{eff}}{U}$ at $\phi = 65°$ was nearly a constant for all values of U. This being the case, a constant value of p can be used. The values of p calculated for each sensor are

Sensor A: $p = 0.76$

Sensor B: $p = 0.80$

Sensor C: $p = 0.78$

Note here that an average value of p must be used in equation 3-11.

Figure 3-5 shows the correlation between the calibration data (for sensor C at U = 16 ft/sec and U = 32 ft/sec) and the $\cos^p\phi$ and K^2 methods with p = 0.78 and K^2 = 0.10. It was observed from this plot that U has little effect on the variation of V_{eff}/U with ϕ. This is most likely due to the relatively large diameter sensors in use here.

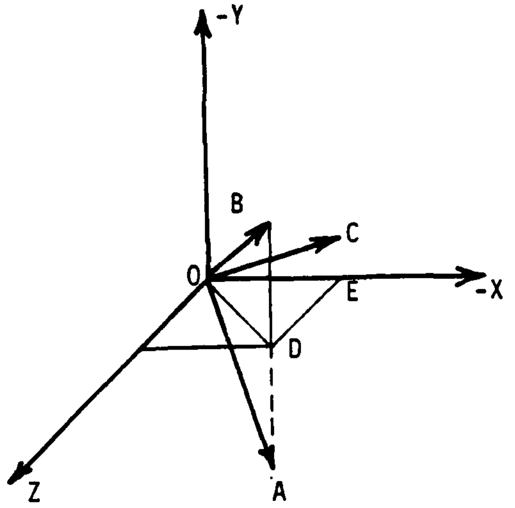


Figure 3-3. Probe Coordinate System Relative to Tunnel Coordinate System

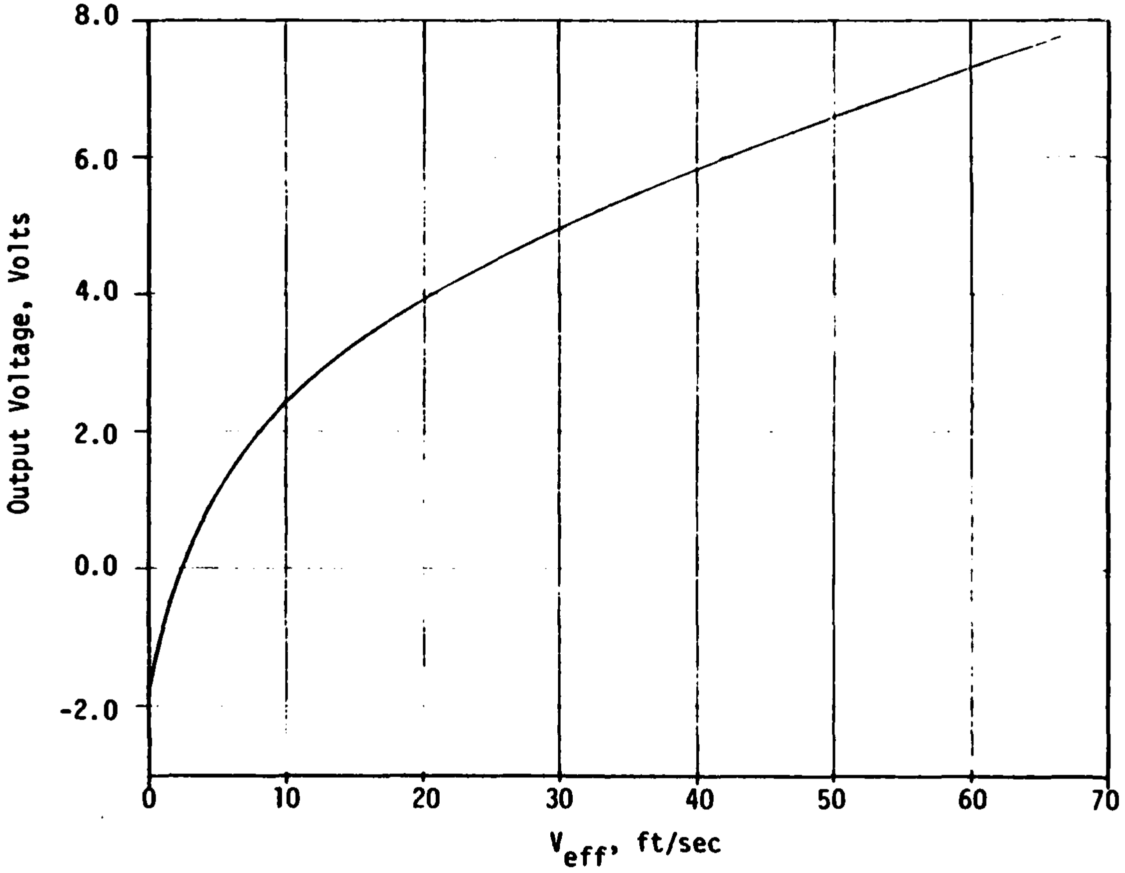


Figure 3-4. Velocity Calibration Curve for Sensor A

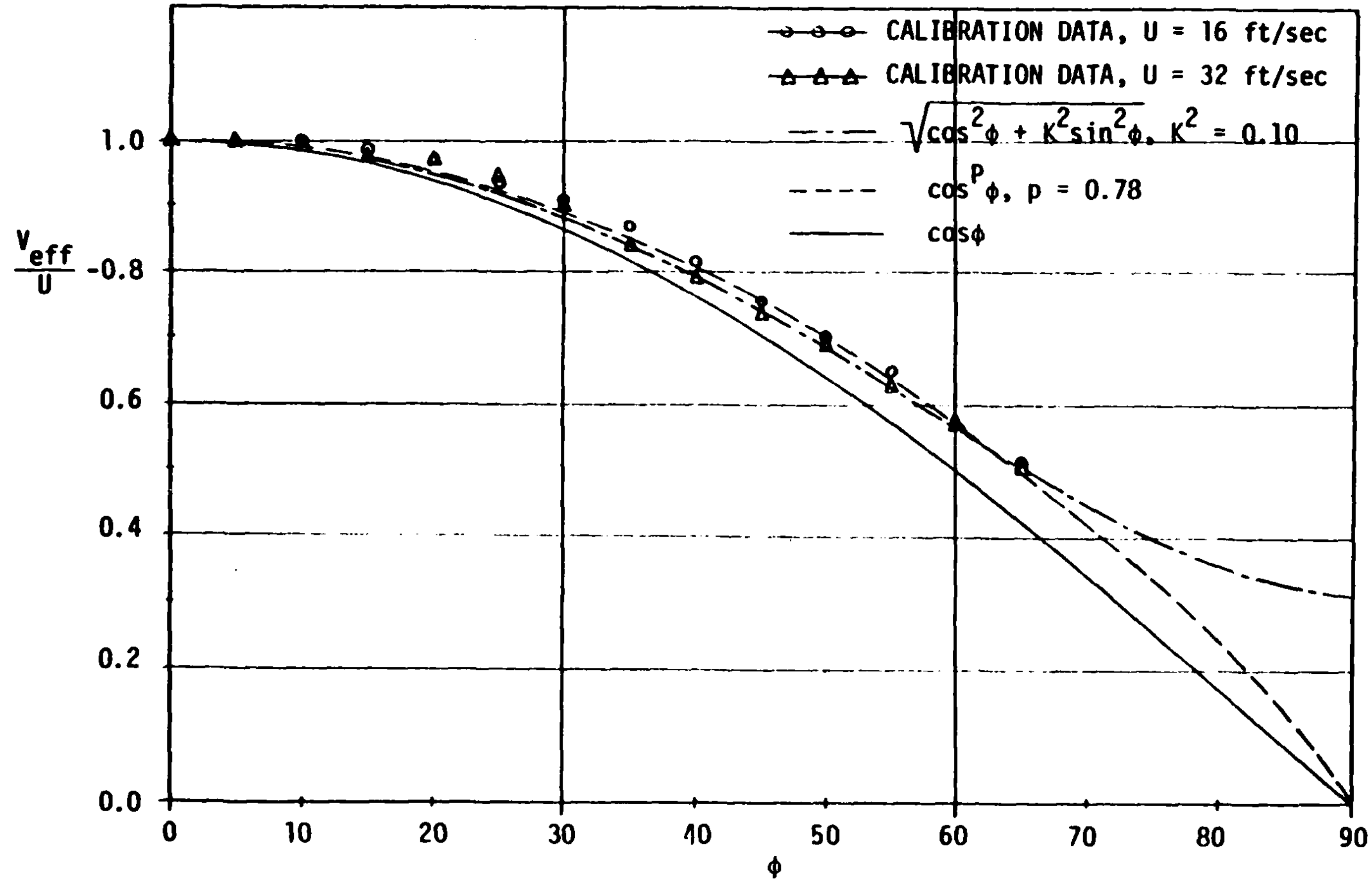


Figure 3-5. Correlation Between Calibration Data and the K^2 and $\cos^P$ Methods

CHAPTER IV

PRESENTATION AND DISCUSSION OF TEST RESULTS

4.1 Test Conditions

The anemometer system of section 2.2 was used to acquire the averaged instantaneous data for an entire case which consists of 1188 points of velocity information. The frequency response of the system as determined by the square wave test, Ref. (3), is approximately 400 Hz.

The parameters of the flight condition were as follows:

Blade Tip Speed, ft/sec	300
Advance Ratio	0.06
Collective Pitch (75%R), degrees	8
Blade Twist, degrees	8
Rotor Shaft Tilt Angle, degrees	8

4.2 Data Presentation

The velocity information is presented here in three forms. Figures 4-1 through 4-3 show the y-z, x-z and x-y vectors respectively for the blade azimuth, $\psi = 355°$. Figures 4-4 through 4-6 show a selected x, y and z plane respectively, for blade angles from 0° to 315° in 45° increments. The data is available in 5° increments but is not presented as such in the interest of brevity. It should be noted here that in Figures 4-1 to 4-6 the ordinate and abscissa scales represent the non-dimensionalized tunnel location and the "+" signs represent the origin of the vectors. Velocity magnitude scaling can be done using the scale provided in the legend of each plot.

The third format used to present the data is that of Figure 4-7. These graphs show the variation of velocity (V_x, V_y and V_z) in one rotor revolution for an individual point in the test volume. A traverse of six points in the z-direction is depicted in Figure 4-7, starting at Z/R = 0.8 and ending at Z/R = -0.2. Table 4-1 gives the non-dimensionalized tunnel location and point number, where the point number is in reference to the 1188 point matrix of the test volume.

Table 4-1. Tunnel Location Reference for Fig. 4-7

POINT #	X/R	Y/R	Z/R
444	-1.07	0.27	0.8
462	-1.07	0.27	0.6
480	-1.07	0.27	0.4
498	-1.07	0.27	0.2
516	-1.07	0.27	0.0
534	-1.07	0.27	-0.2

The vector plots of Figures 4-1 through 4-6 were made using the program VECTOR.FTN whereas the plots of Figure 4-7 were made using the program EZPLOT.FTN in conjunction with the system graphing routine called EASY GRAPHING, see Appendix C. Also in Appendix C are the tabular data for the plots of Figure 4-7.

The data of the azimuth-varying vector plots of Figures 4-4 through 4-6 are also available in an animated film version. The filming technique is presented in Appendix D.

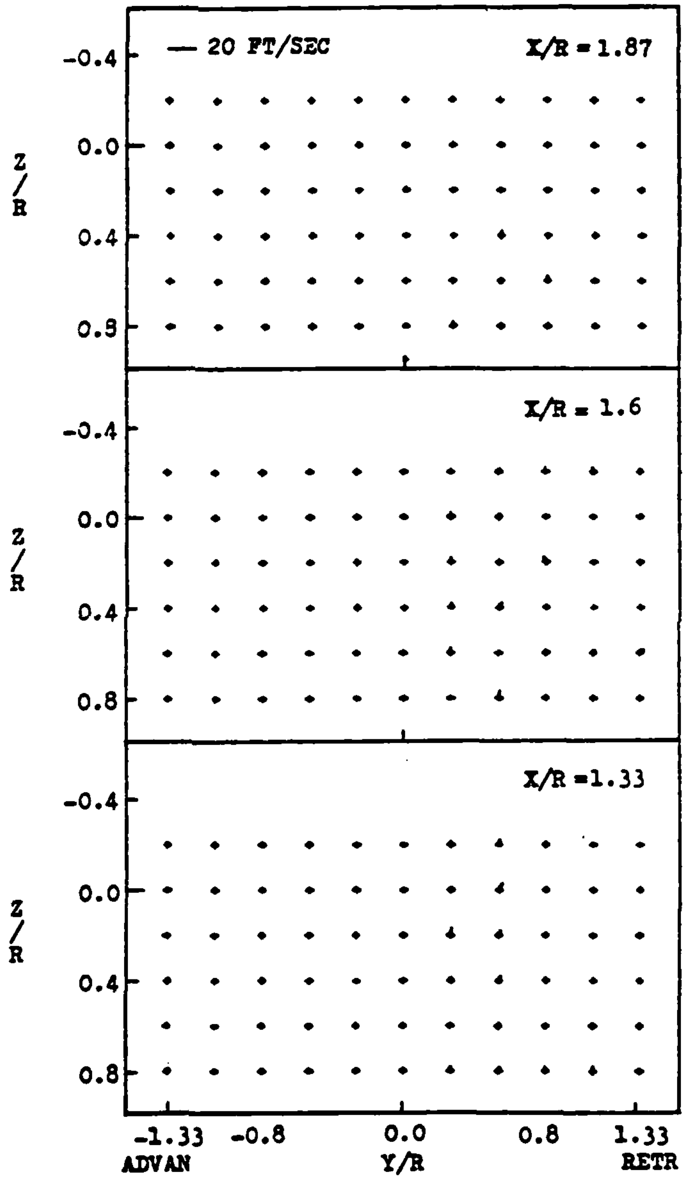


Figure 4-1. Y-Z Vectors Looking Upstream, ψ = 355°

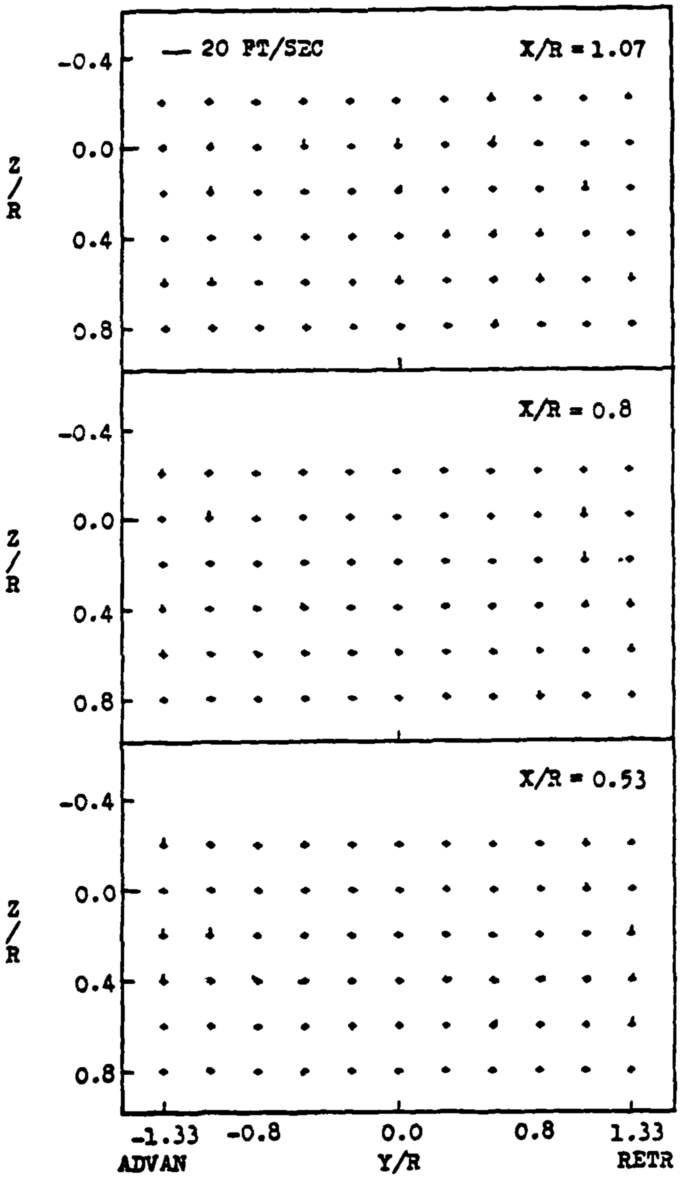


Figure 4-1. continued

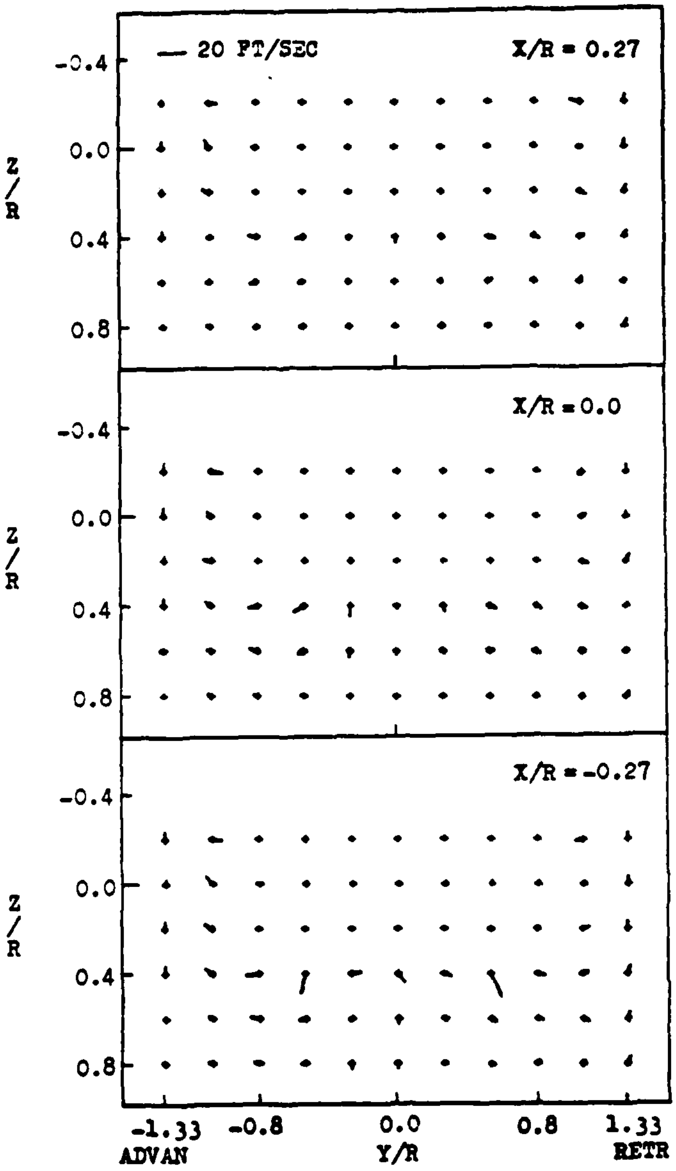


Figure 4-1. continued

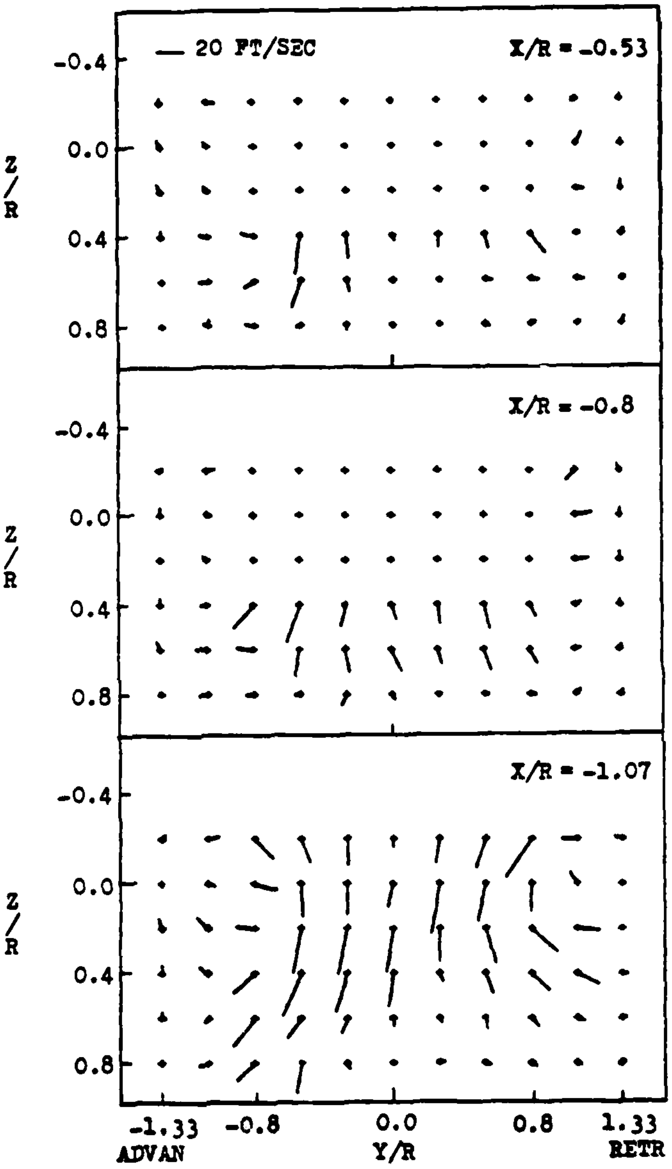


Figure 4-1. continued

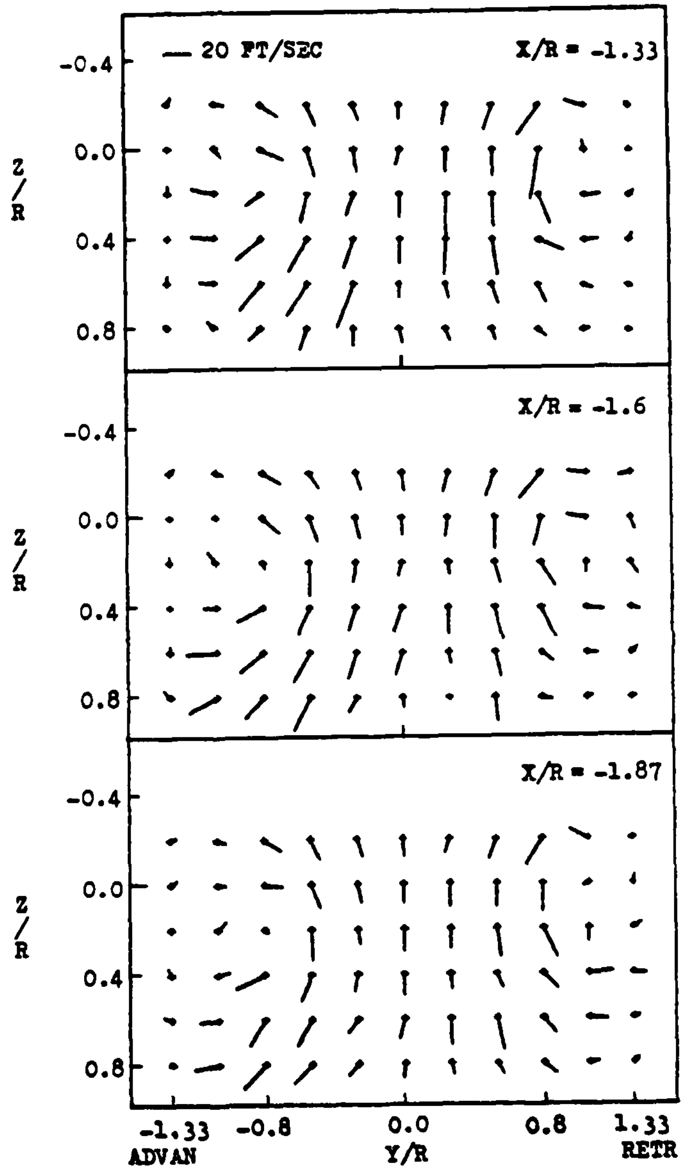


Figure 4-1. continued

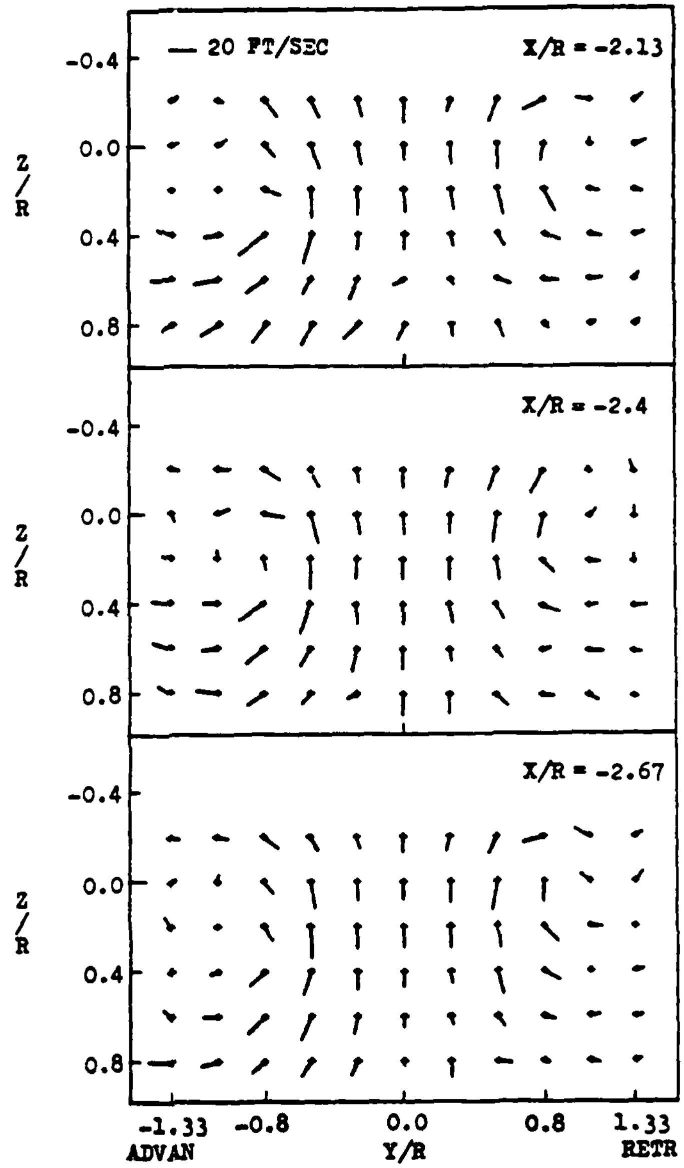


Figure 4-1. concluded

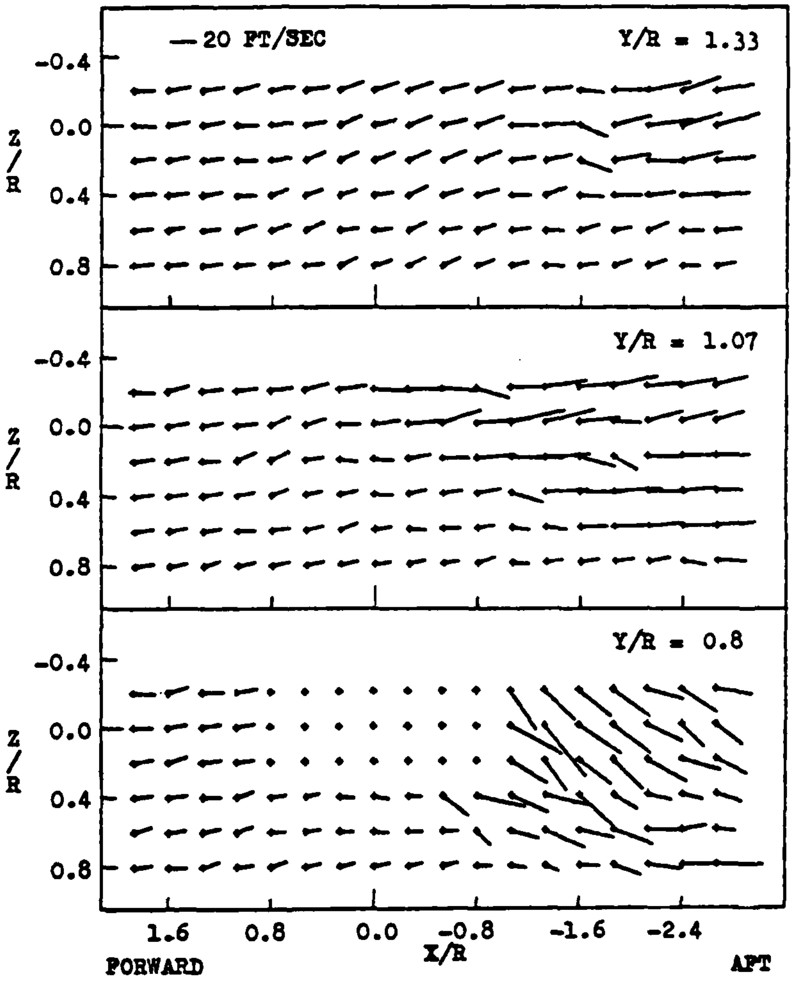


Figure 4-2. X-Z Vectors for ψ = 355°

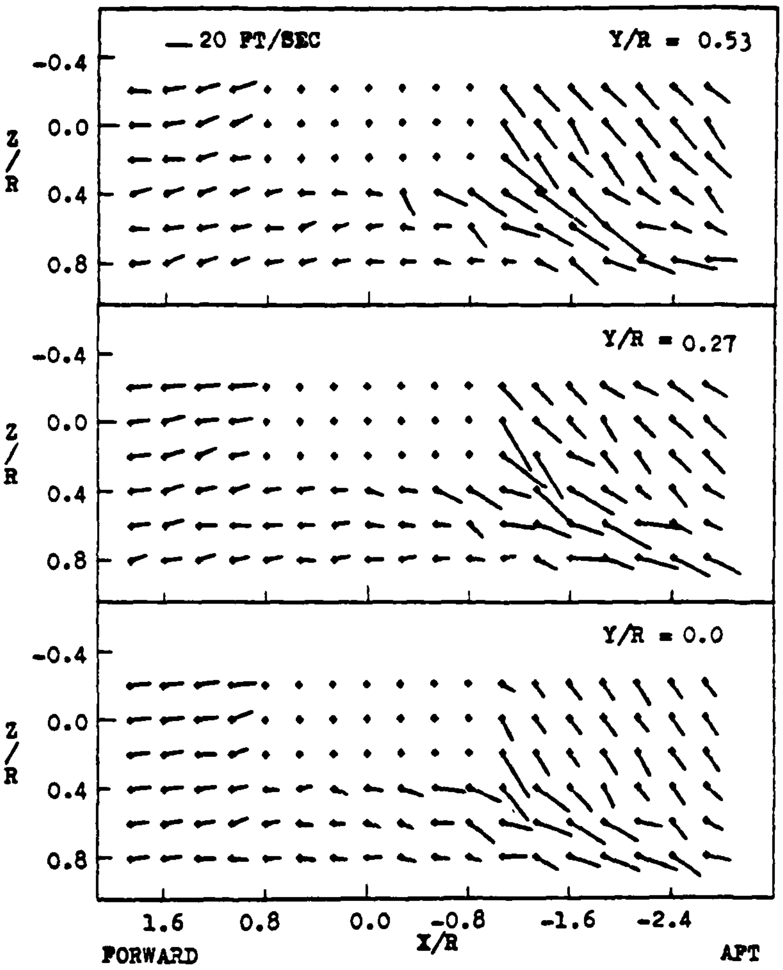


Figure 4-2. continued

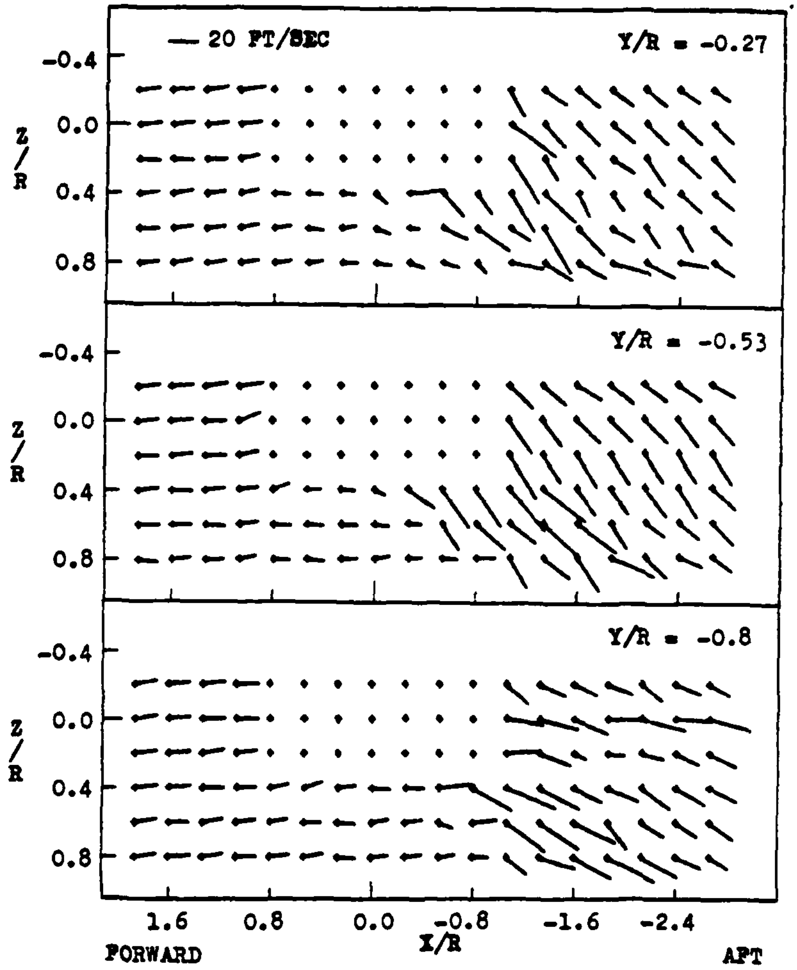


Figure 4-2. continued

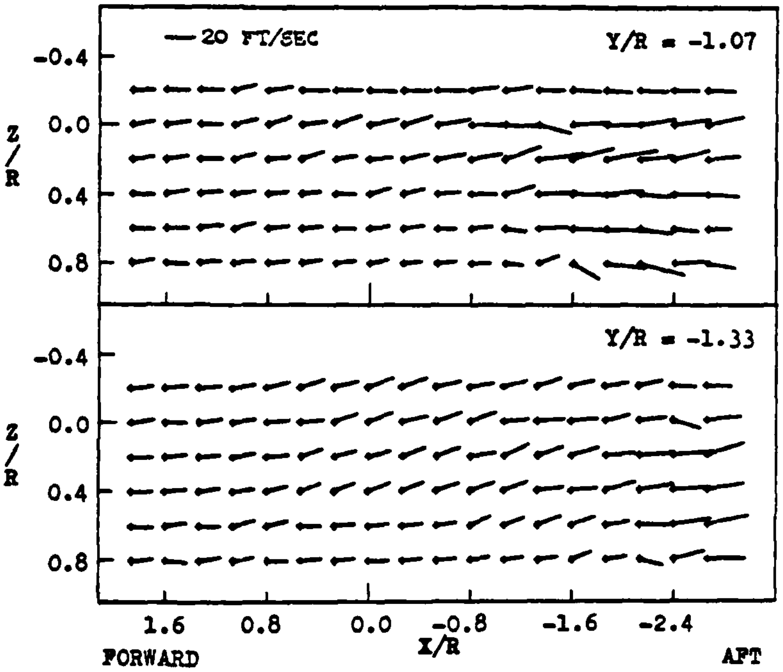


Figure 4-2. concluded

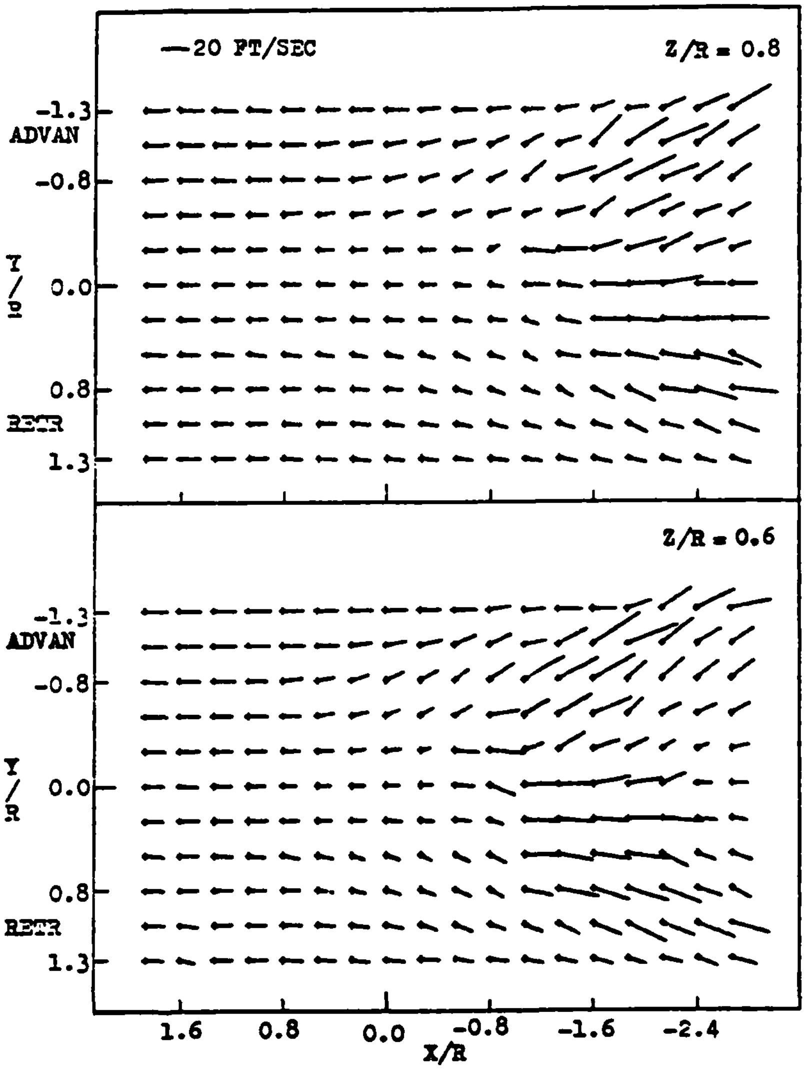


Figure 4-3. X-Y Vectors Viewed from Below the Rotor, ψ = 355°

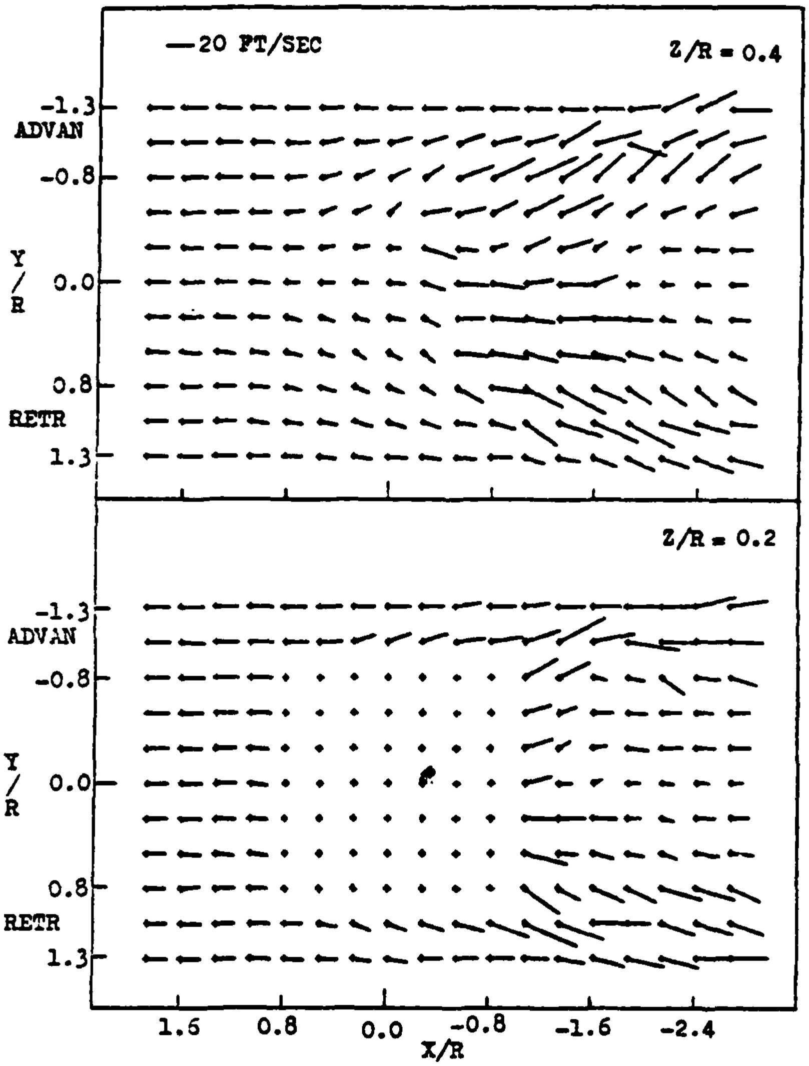


Figure 4-3. continued

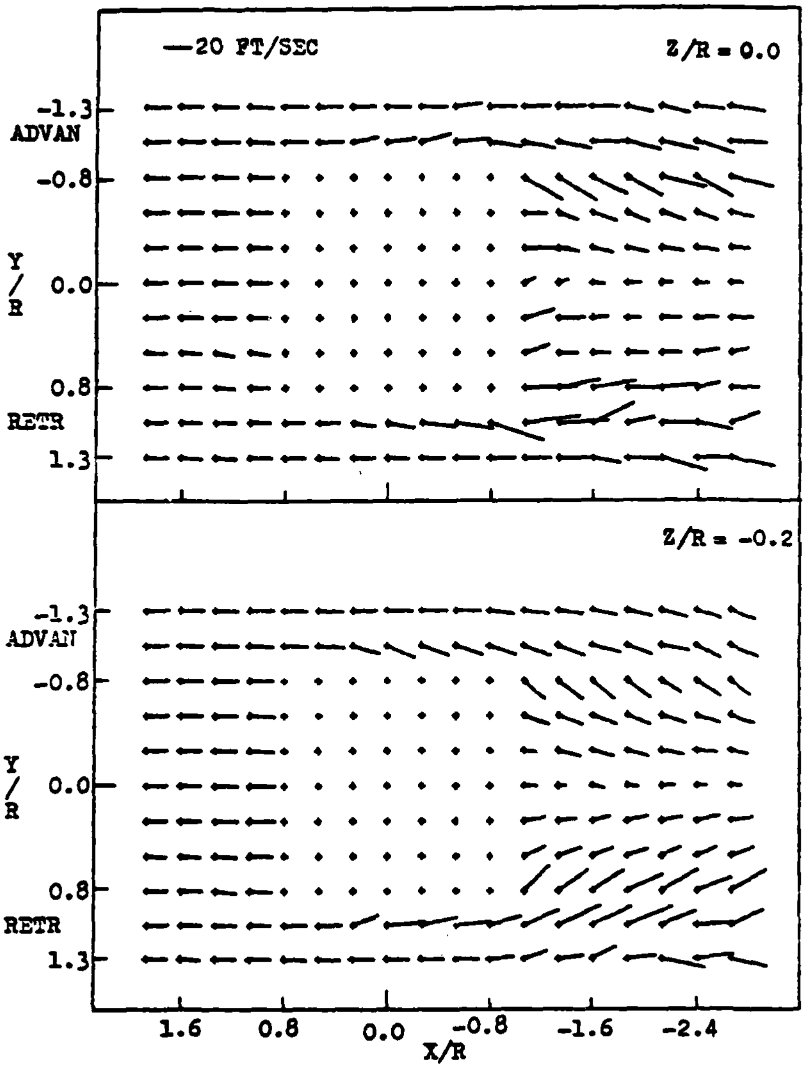


Figure 4-3. continued

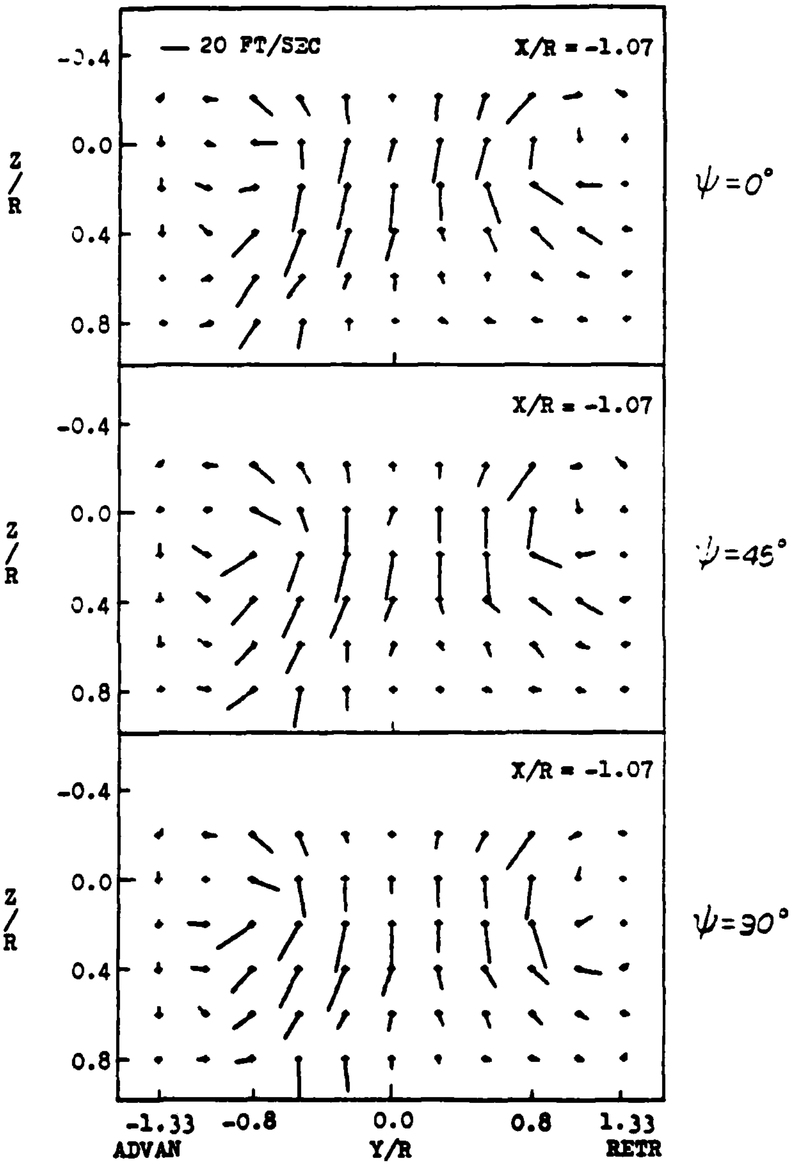


Figure 4-4. Y-Z Vectors for X/R = -1.07 at Several Blade Angles

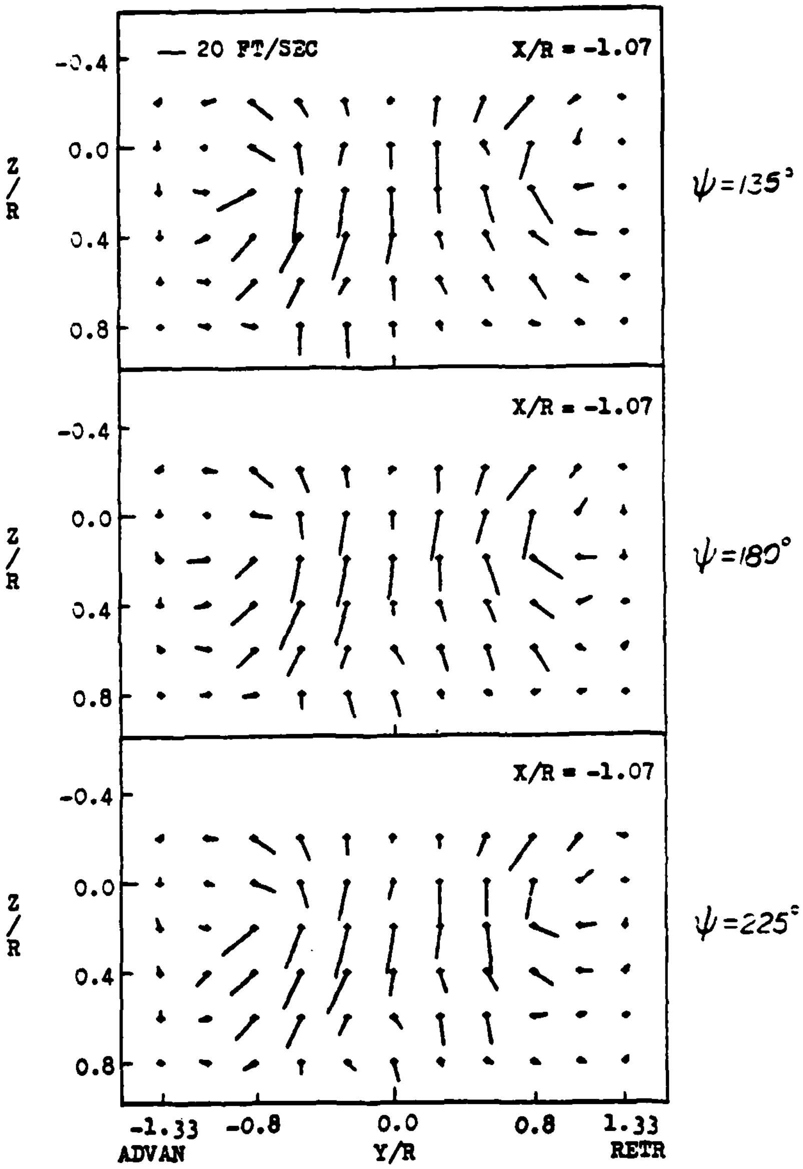


Figure 4-4. continued

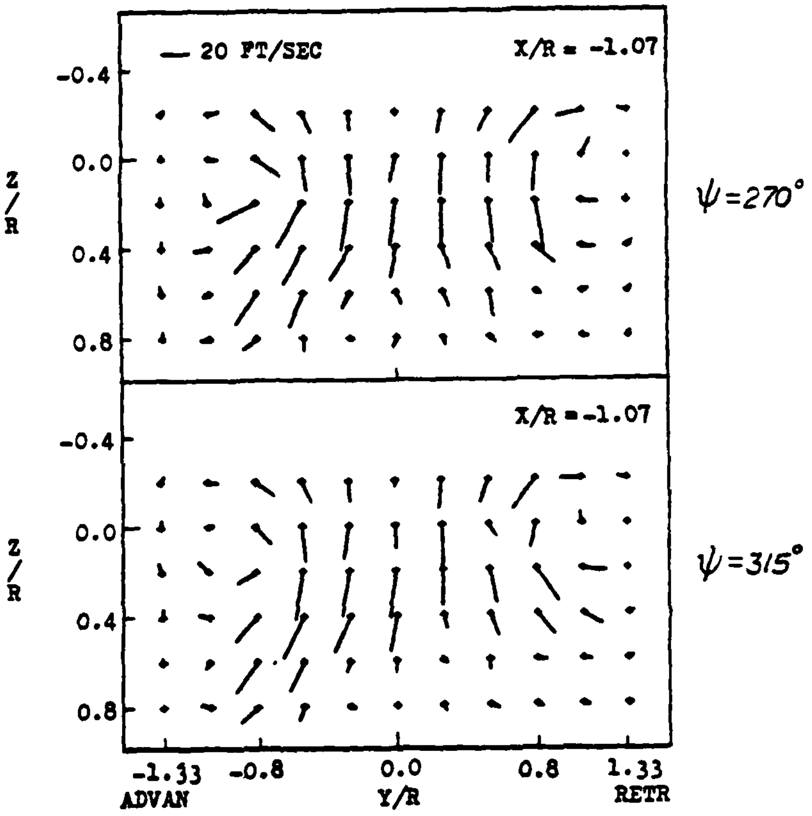


Figure 4-4. concluded

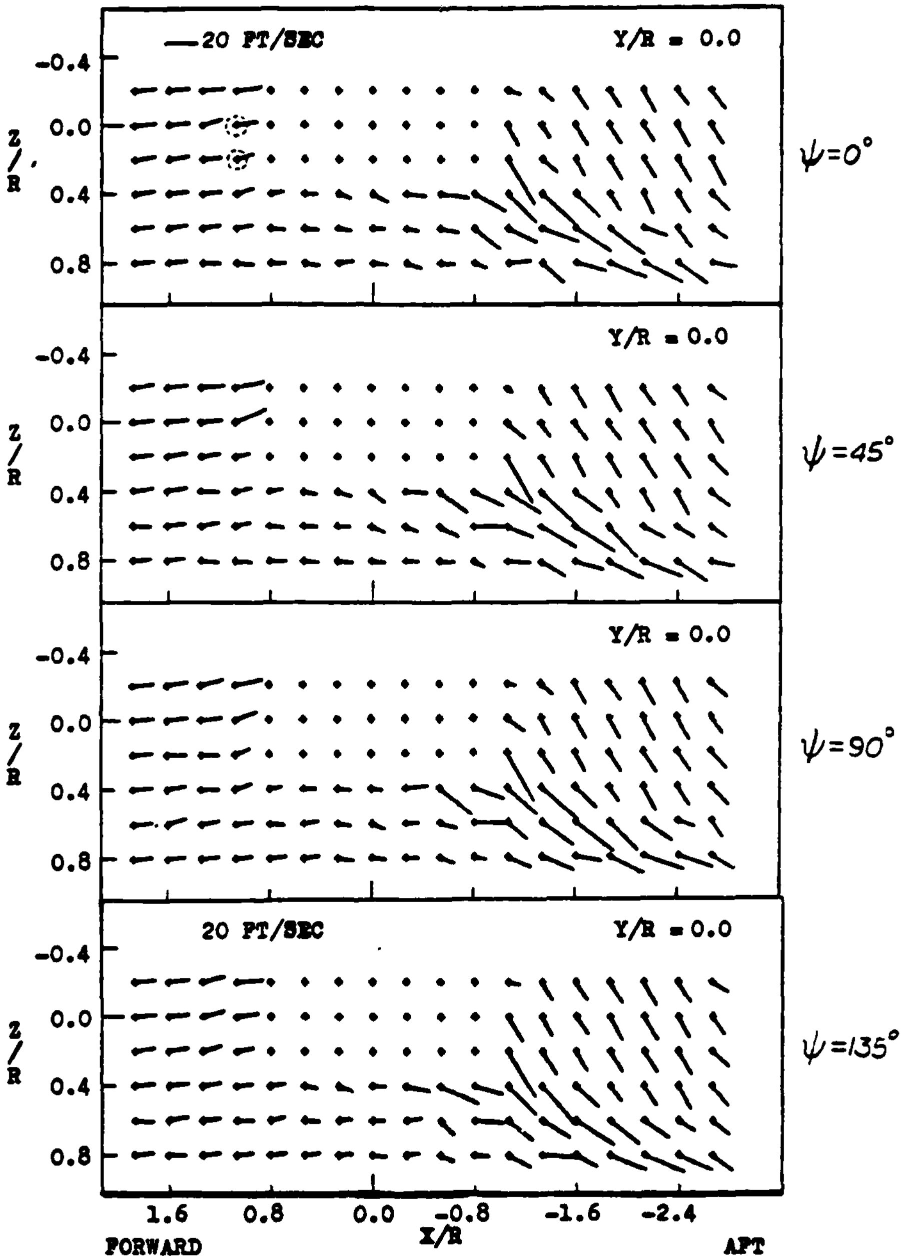


Figure 4-5. X-Z Vectors for Y/R = 0.0 at Several Blade Angles

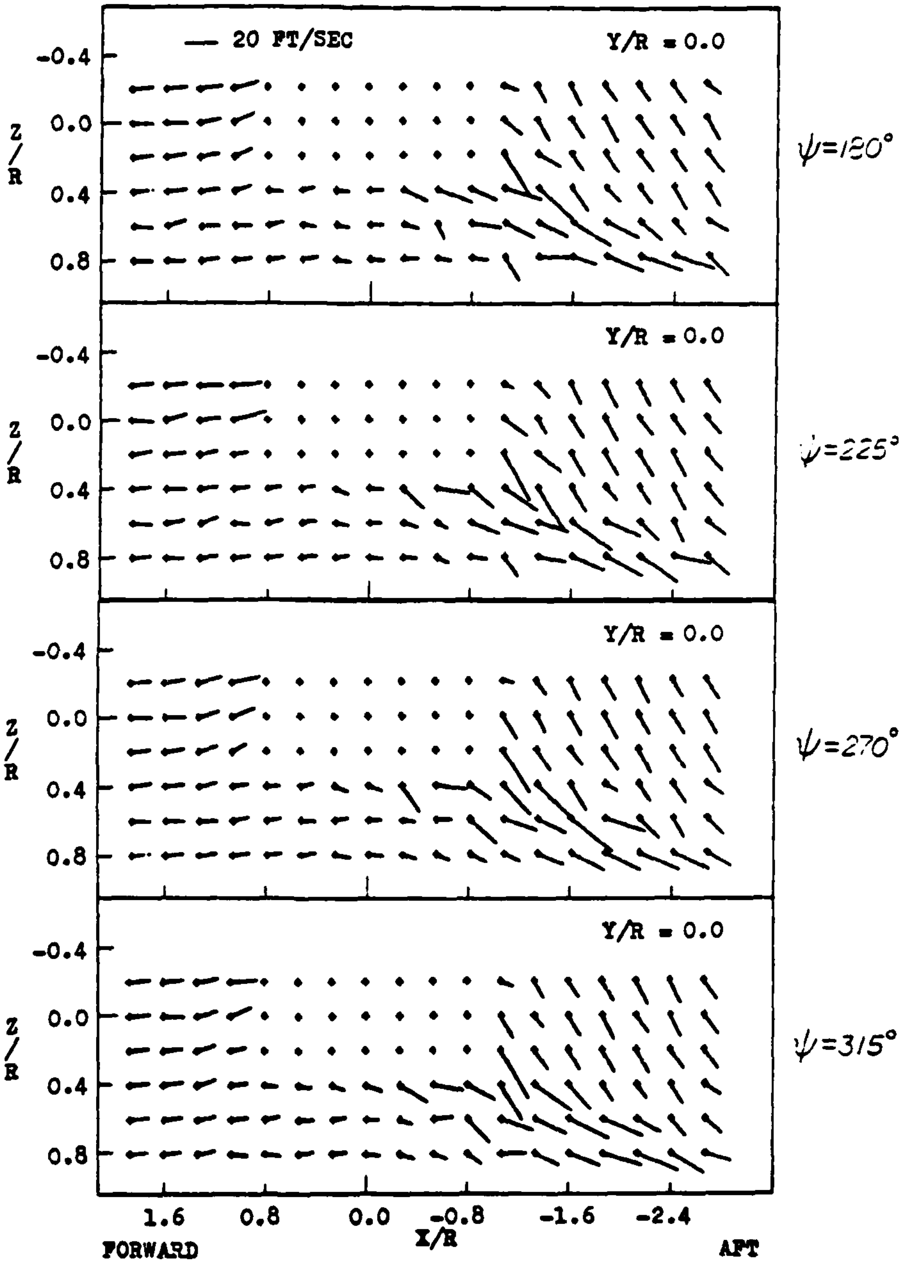


Figure 4-5. concluded

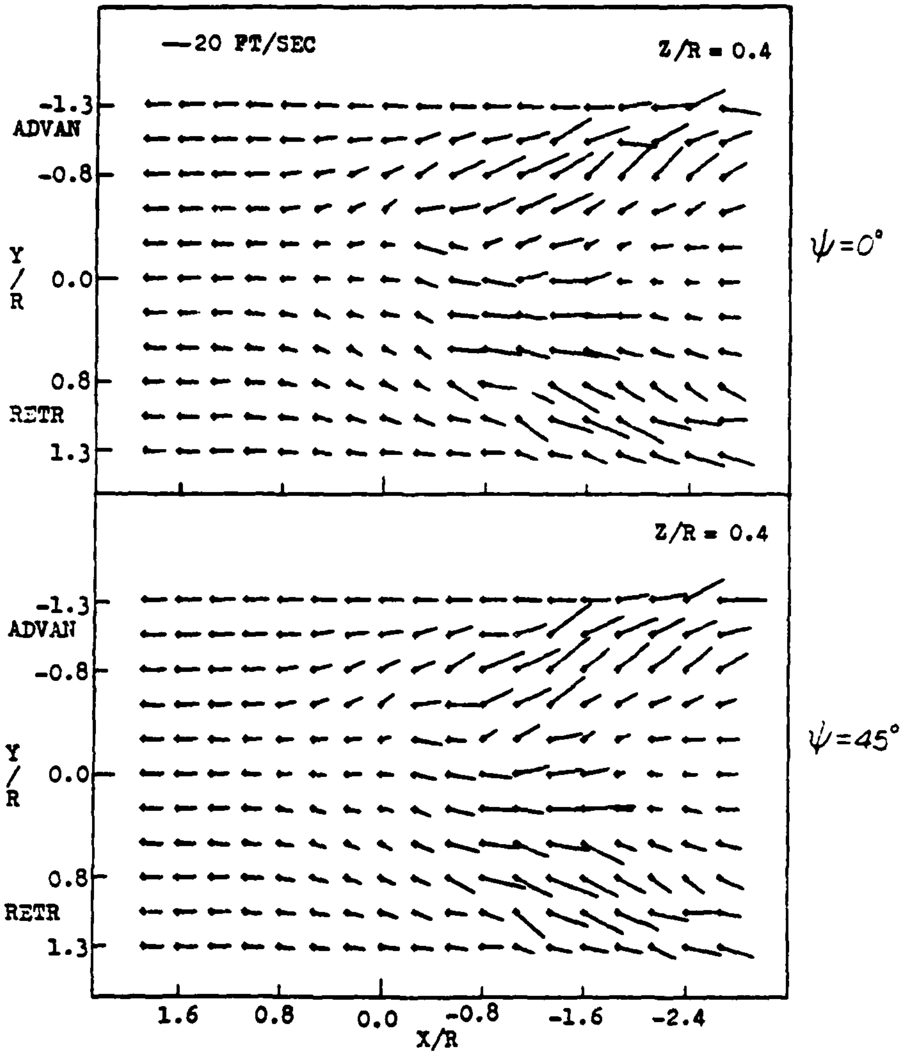


Figure 4-6. X-Y Vectors for Z/R = 0.4 for Several Blade Angles

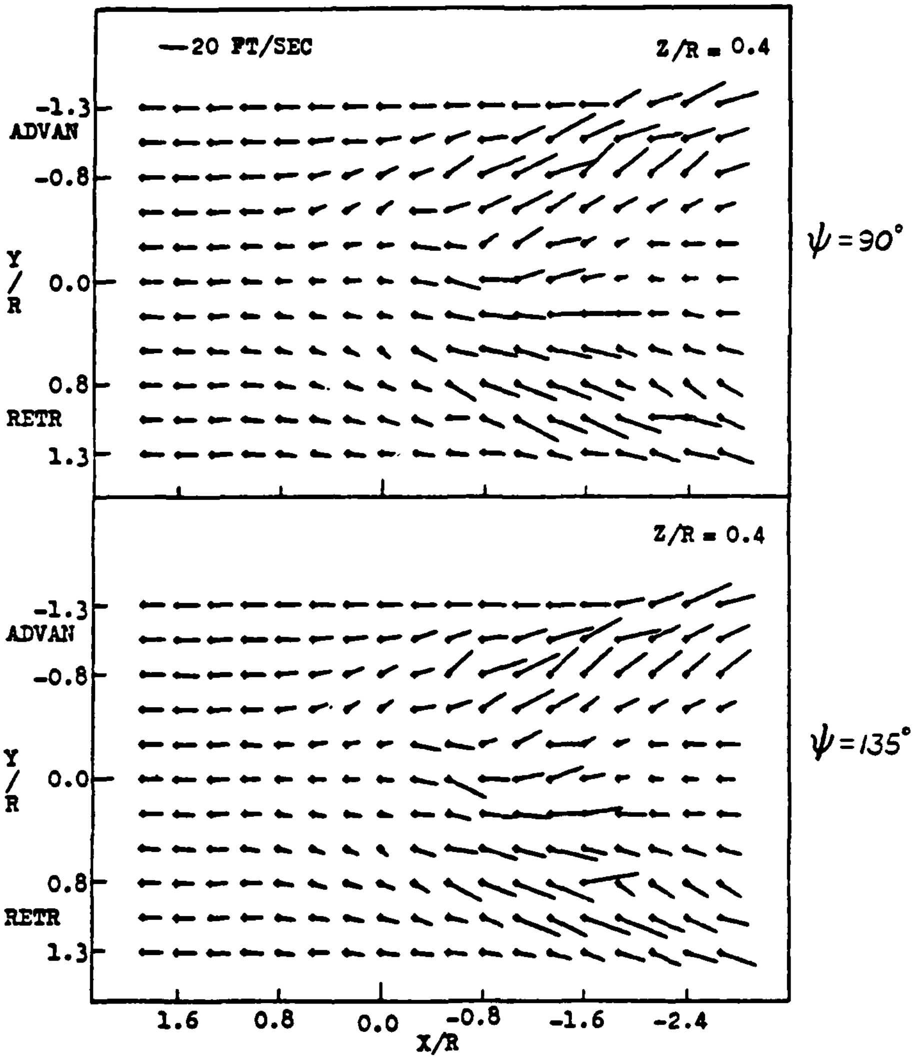


Figure 4-6, continued

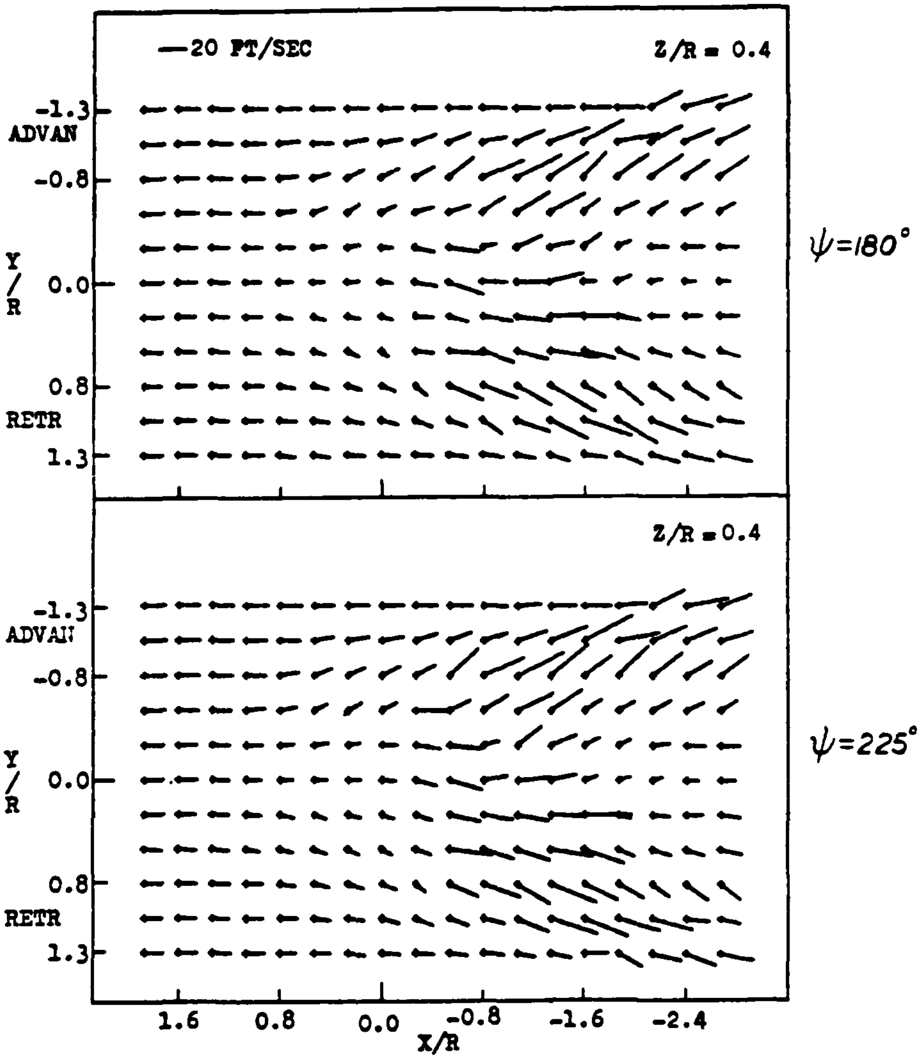


Figure 4-6. continued

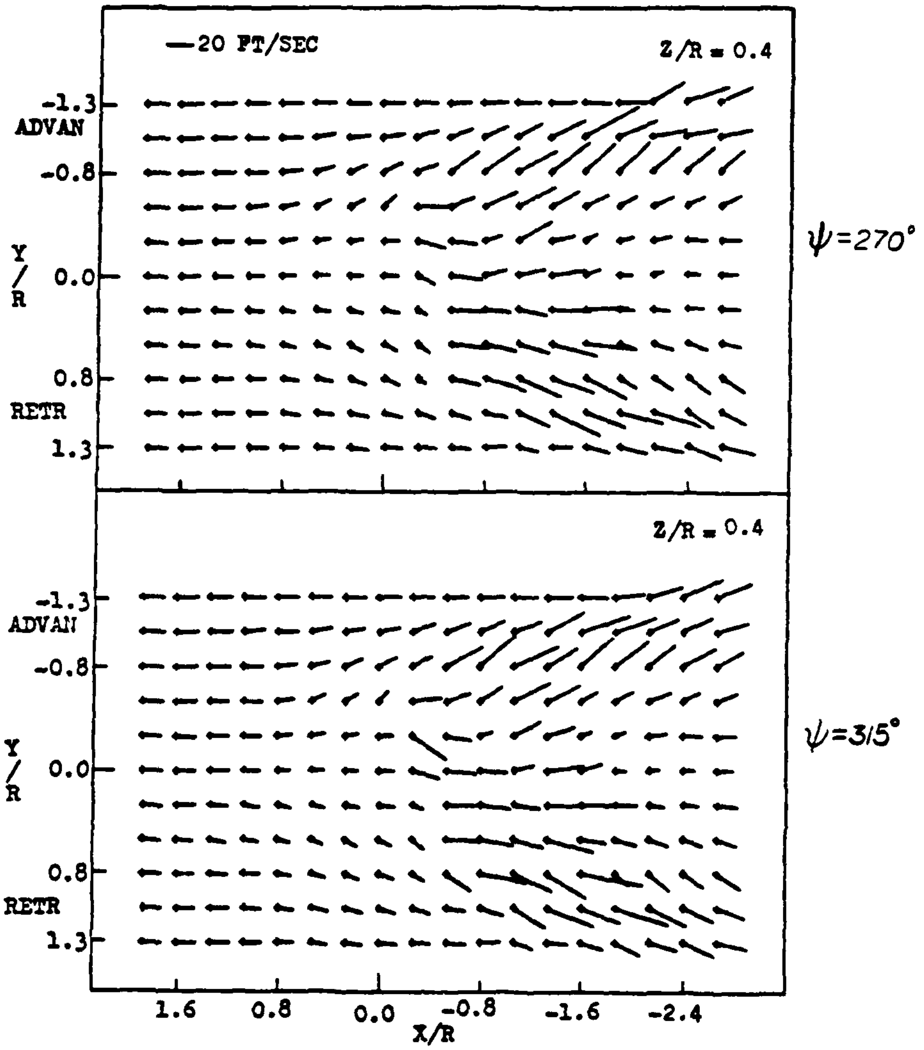


Figure 4-6. concluded

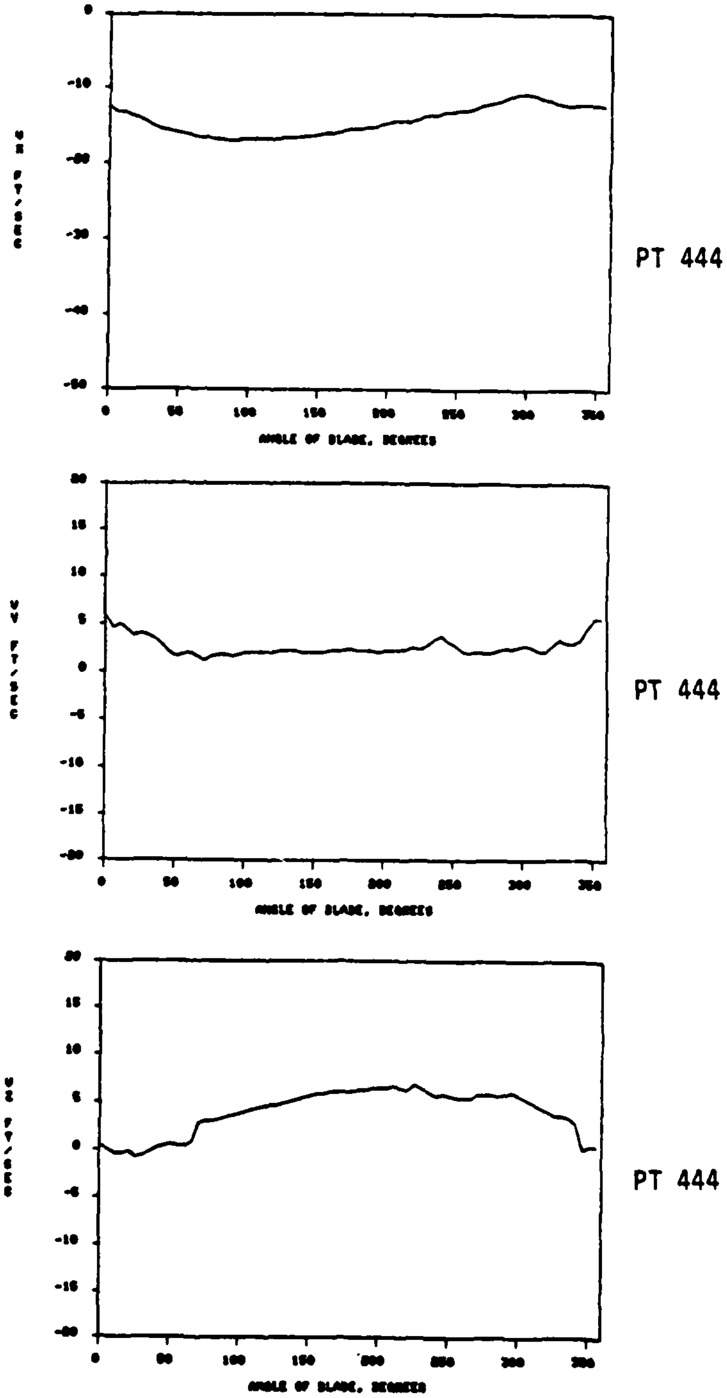


Figure 4-7. Azimuth Varying Velocities for 1 Blade Revolution
Z Traverse at Y/R = 0.27 and X/R = -1.07

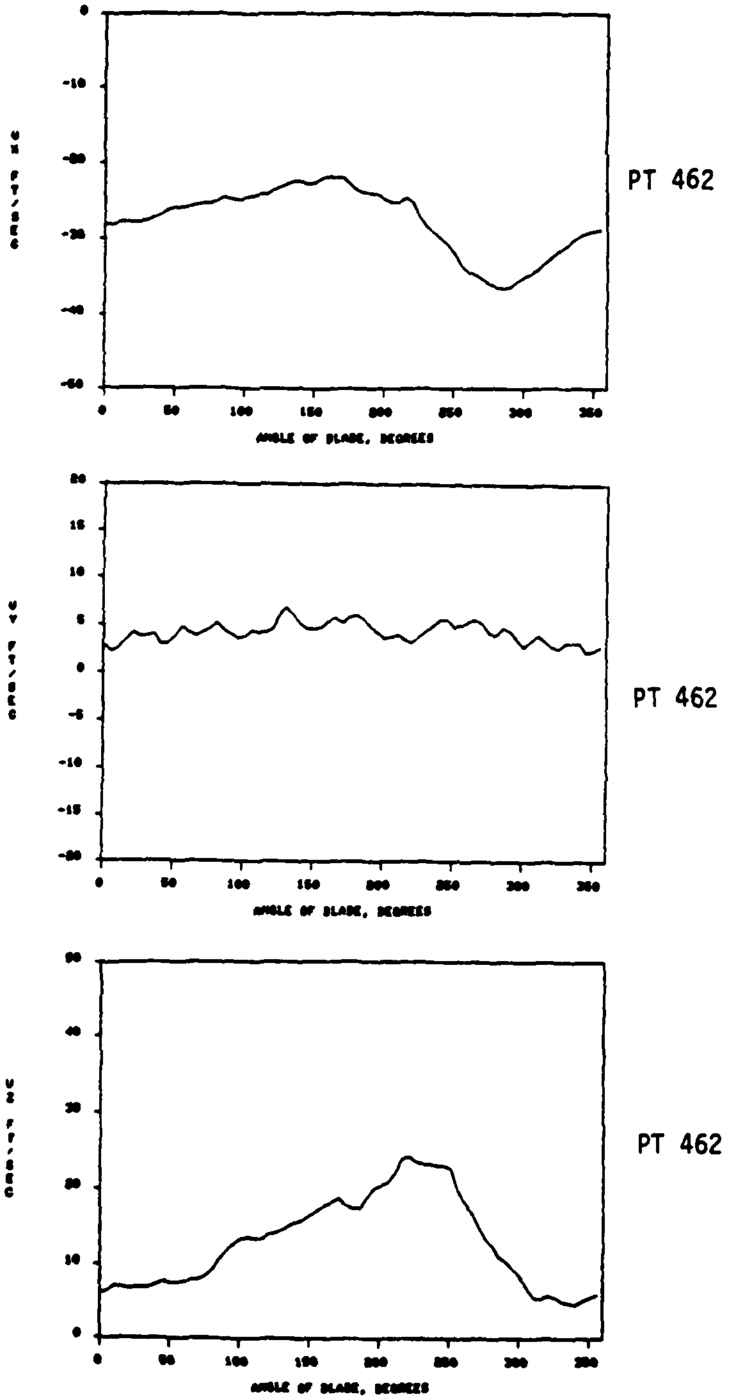


Figure 4-7. continued

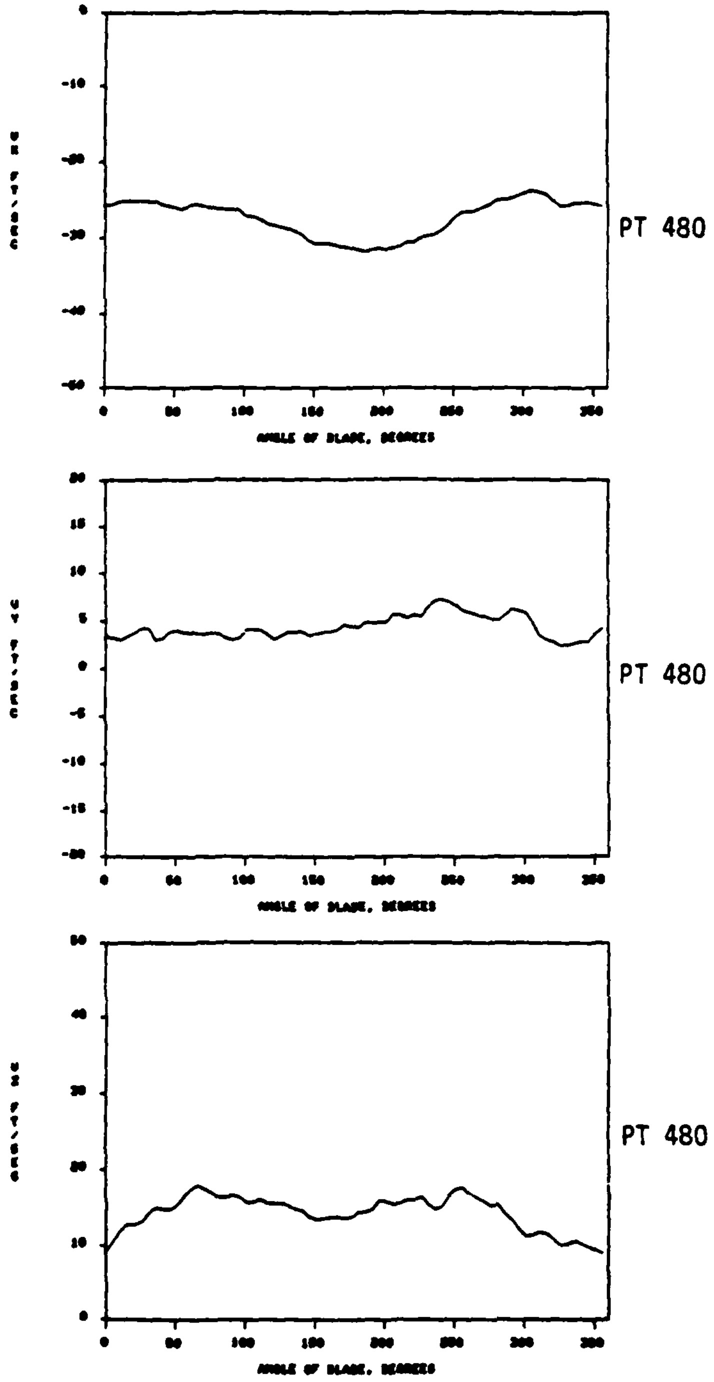


Figure 4-7. continued

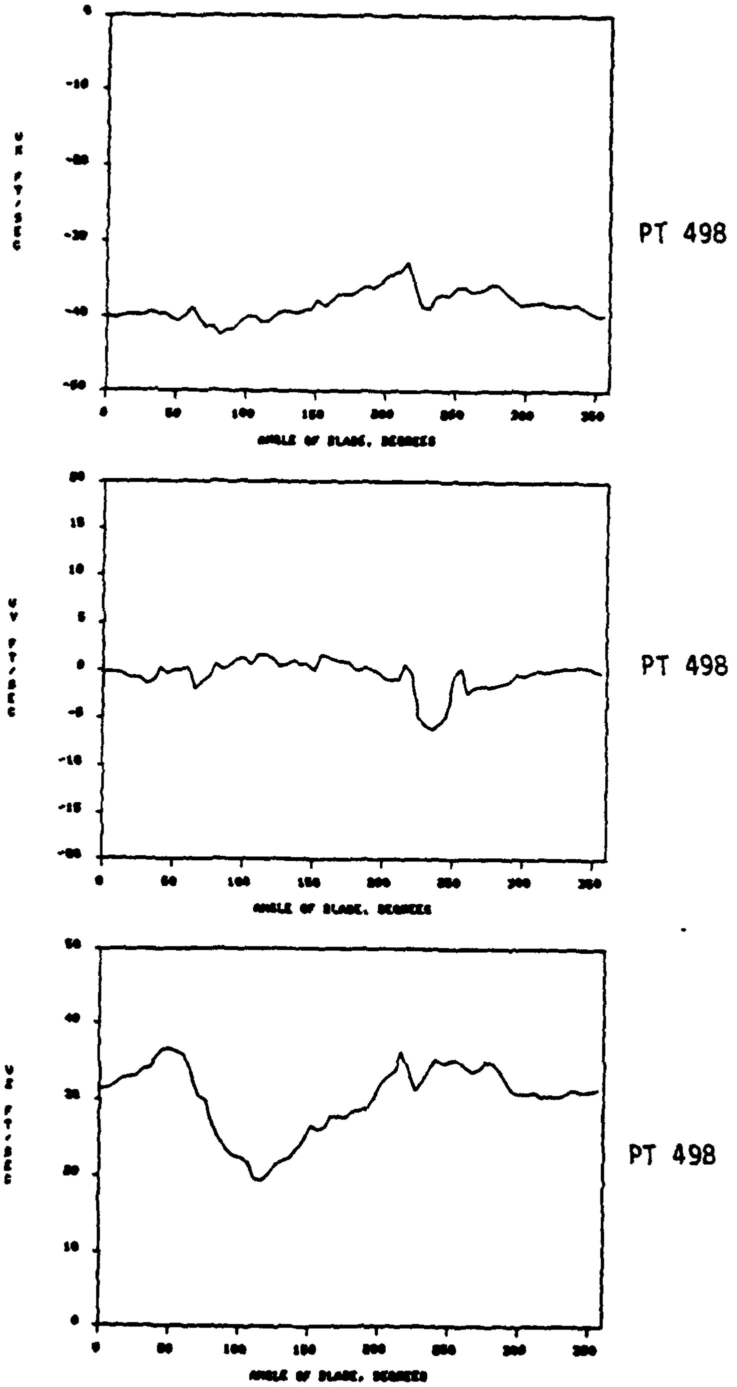


Figure 4-7. continued

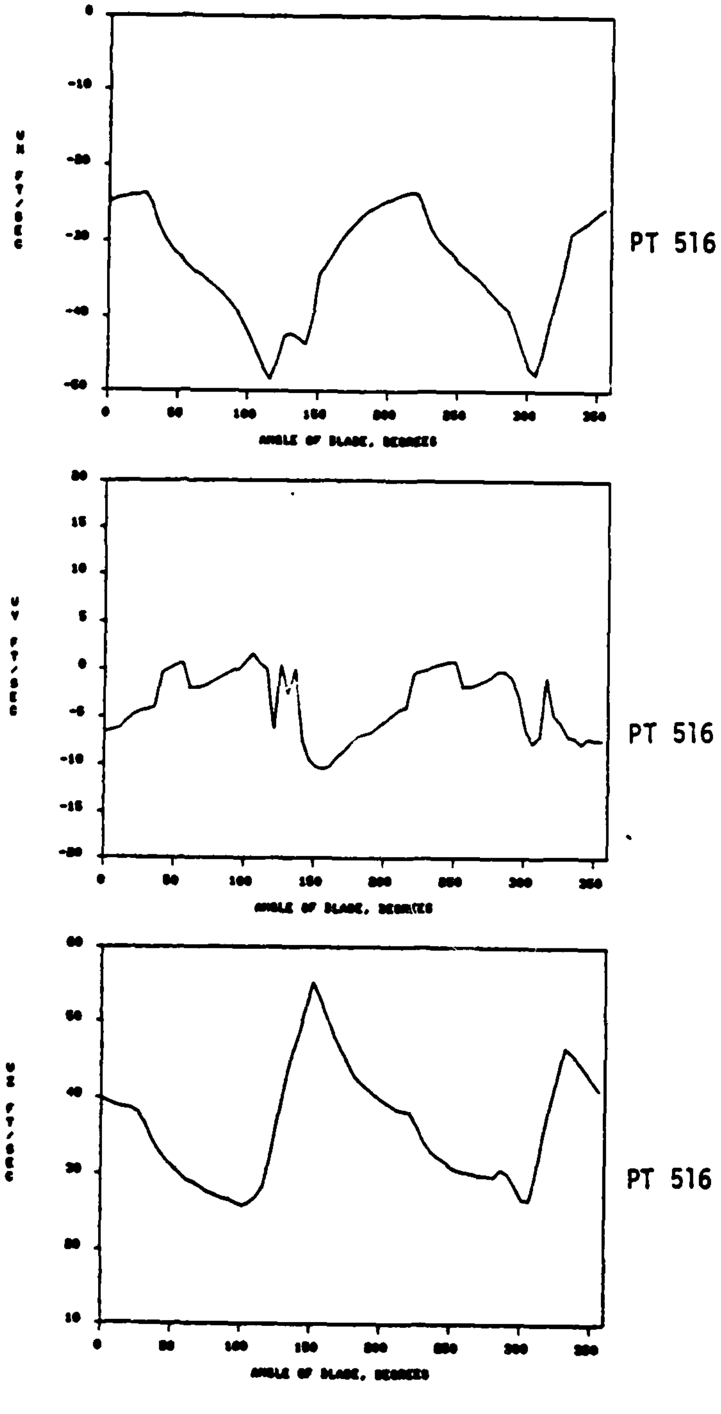


Figure 4-7. continued

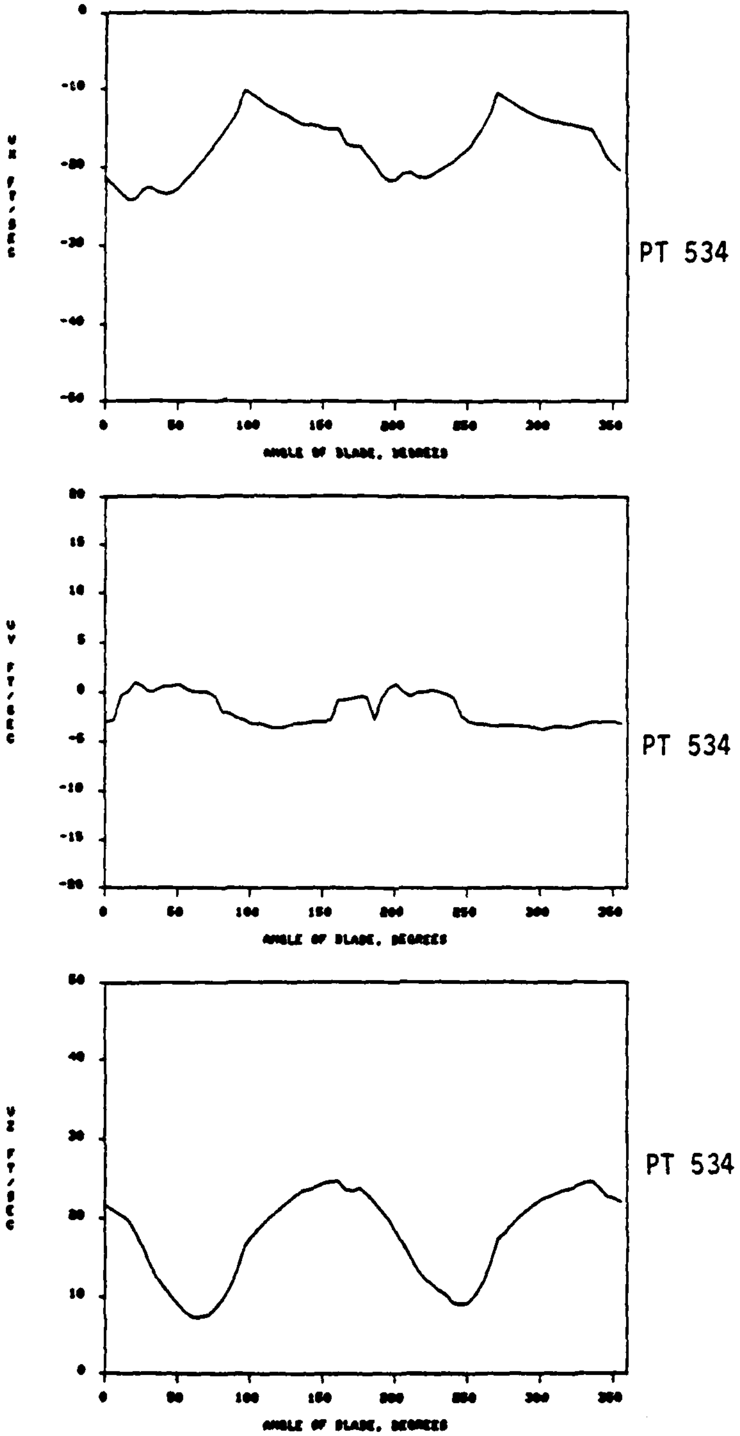


Figure 4-7. concluded

4.3 Discussion of Results

As can be seen by the representative data in Figures 4-1 through 4-7, there is a vast amount of information available for analysis. Ideally, the data should pertain to a helicopter in level unaccelerated forward flight. This condition exists when the thrust vector is coincident with the rotor shaft axis. However, no provision was made to ensure this condition. Consequently, no specific flight attitude can be presumed. Despite this limitation, the data presented demonstrates the feasibility of obtaining qualitative and quantitative averaged instantaneous rotor wake velocity information.

Of particular interest in the transition flight regime is the nature of the rollup phenomenon of the trailing vortices. This is shown clearly in Figure 4-1 beginning dt X/R = 0.27 suggesting that the vorticity sheet rolls up very quickly. Note also that this activity is occurring first on the advancing side. At location X/R = -1.07 the rollup is fully developed and a very strong downwash velocity is exhibited. Once again the plots fail to show axisymmetric flow patterns in that the downwash velocities on the advancing side are greater in magnitude than on the retreating side. Nonuniform induced velocities suggest a nonuniform disk loading. Nonuniform disk loading is expected for a trimmed rotor, but since the rotor has not been trimmed it is impossible to attribute the nonuniformity to a specific cause. The anticipated downward and aft motion of the trailing vortex center can be seen when comparing the apparent center on the advancing side for X/R = -1.07 and X/R = -2.67. This downward progression is not well defined on the retreating side.

The azimuth-varying Y-Z vector plots of Figure 4-4 do not show obvious changes in all vectors from one blade position to the next, however upon closer inspection, some specific vectors are widely varying. Take as an example the vector at the point Y/R = 0.53 and Z/R = 0.0. During the blade revolution its magnitude changes by a factor of 4 and its direction changes ± 20° from the vertical direction.

The lateral progression of X-Z planes (Fig. 4-2) from the retreating to advancing sides shows the expected strong downwash patterns typical of flows for low advance ratios. The downwash is most pronounced at the Y/R = ± 0.53 locations. At the location Y/R = 0.0 the effect of the tip vortices rotating in the X-Z plane (this will be the case at $\psi = 0°$) is reflected in the behavior of the X-component. The counterclockwise rotation of a vortex in the region just below and just aft of the rotor will add to the stream velocity in the vicinity of Z/R = 0.4 to 0.8 and subtract from it in the 0.0 to -0.2 region. Beyond the blade tip path in both directions along the lateral axis and just forward of the rotor these same tip vortices cause a slight upflow which can be seen clearly in Fig. 4-2 at locations Y/R = ± 1.33 and Y/R = 0.0.

A very obvious indication of the azimuth-wise variation of the flow can be seen in Fig. 4-5. Bear in mind that at $\psi = 0°$ the blades are in the plane of the page and rotating clockwise as viewed from above the rotor disk. Noting the two vectors with circled origins in the $\psi = 0°$ plot and tracking them through the revolution it can be seen that the X-component of these two vectors in the $\psi = 45°$ plot now show the influence of a clockwise rotating tip vortex. At ψ = 135° the X-components are nearly equal again, followed by a nearly identical repetition of the vortex influence at ψ = 225°.

The X-Y plots of Fig. 4-3 show a near total negation of the stream velocity at Y/R = 0.0, Z/R = -0.2 and X/R = -1.07 to -2.4, again showing the tip vortex flow.

The azimuth-varying velocities, VX, VY and VZ of Fig. 4-7 substantiates what has been shown in Figs. 4-1 to 4-6. Note the clear two per revolution frequency appearing in the X and Z velocities for points 516 and 534. Moving further below the rotor to points 498, 480, 462 and 444 shows a degeneration of this frequency probably caused by the probe being more removed from intense vortex activity.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

In its present form, the entire system, including the computer programs, is an invaluable tool in the data acquisition and presentation process. The use of the Mechanical Engineering Department minicomputer and A/D converter greatly expanded the capability and efficiency of the system. The time needed to take the data for the case was approximately 8 hours. This is an improvement over the previous method which took 20 hours.

The calibration process is currently very cumbersome and time consuming. It is suggested that the A/D converter and the minicomputer be used instead of the current method of using a digital voltmeter and the IBM 470 batch system. This would eliminate hours of work with computer cards. The calibration results could be stored in a direct access file which would subsequently be read when the data taking program is run (i.e. COSPX.FTN). This would eliminate the need to enter the constants of the polynomials by hand.

Also of interest in terms of streamlining the tunnel operators job would be to link the traverse mechanism to the minicomputer which would automatically control the position of the probe in the tunnel. A total redesign of the traverse mechanism and the purchase of a microprocessor would be necessary. The microprocessor would allow the flexibility of programmability in the event the test volume should be changed. In contrast, a hard wired circuit to perform the same task would require extensive rework if any test volume changes are made. The new traverse mechanism design should allow complete motion from the floor to the ceiling of the tunnel without the currently required operator assistance.

CONCLUSIONS

The filmed animation of the rotor flowfield for a single rotor revolution demonstrates the feasibility of this format for flow visualization. The overall trends of the data show agreement with time-averaged data and smoke visualization data of the literature, Ref. (5). Expected phenomenon such as the rollup of the trailing vortex in the Y-Z planes and the strong downwash velocities just aft of the rotor are both typical of the helicopter rotor in forward flight.

Although the instantaneous nature of the measured velocities in the x, y and z directions is readily apparent in the azimuthwise plots of Figure 4-7, the vector plots, with the exception of a few isolated points, do not show this same degree of marked change. Specifically, the two per rev signal, which is quite pronounced at point 516 in the X and Z directions (Figure 4-7) due to the close proximity of the probe to the rotor tip path plane, undergoes only gradual changes when viewed in the X-Z plane of vectors. For instance, the change in V_X from a blade angle of $\psi = 50°$ to $\psi = 100°$ is about -10 ft/sec. The change in V_Z for this same interval is -5 ft/sec. In vector form the magnitude of the resultant changes from 44 ft/sec to 49 ft/sec and the angle change is about 15°. With the vector shown every 5° this change takes place over a span of 10 pictures between $\psi = 50°$ and $\psi = 100°$. The difference between the two different representations is further amplified by the expanded scale of the V_x, V_y, V_z plots of Figure 4-7 as opposed to the relatively small scale of the vector plots of Figures 4-1 to 4-6. Therefore the azimuth varying plots of V_x, V_y and V_z present the greatest amount of information about the instantaneous velocities.

REFERENCES

1. Champagne, F. H., C. A. Sliecher, O. H. Wehumann, "Turbulence Measurement with Inclined Hot Wires," Journal of Fluid Mechanics. 1967, Vol. 28, part 1, pp. 153-182.

2. Doebelin, E. O., Measurement Systems. New York, McGraw-Hill, 1975.

3. Freymuth, P., "Feedback Control Theory for Constant Temperature Hot-Wire Anemometers," Review of Scientific Instruments, Vol. 38, No. 5, May 1967.

4. Freymuth, P., "Noise in Hot-Wire Anemometers," Review of Scientific Instruments, Vol. 39, No. 4, April 1968.

5. Heyson, H. H. and S. Katzoff, "Induced Velocities Near a Lifting Rotor with Nonuniform Disk Loading," NACA Report 1319, 1957.

6. Horak, D., "Development of an Instrumentation System to Measure the Velocities in the Wake of a Rotor," M.Sc. Thesis, The Ohio State University, 1977.

7. Jones, J. W., H. R. Velkoff, D. R. Mayhew, S. A. Korpela, "A Study of Correlation Parameters Useful for Hot-Wire Anemometry in Yawed Flow," The Ohio State University Research Foundation, RF Project 760069/783375, 1974.

8. Terkel, H., "Nonsteady Velocity Measurement of the Wake of a Helicopter Rotor in Low Advance Ratios," M.Sc. Thesis, The Ohio State University, 1980.

9. Velkoff, H. R., D. Horak, "Rotor Wake Measurements at Very Low Advance Ratios," 35th Annual Forum of the American Helicopter Society, Paper No. 79-G, Washington, D.C., May 1979.

10. Velkoff, H. R., H. Terkel, Fu-Kuo Shaio, Effect of Changing Rotor Parameters on Rotor Wake Velocities at Very Low Advance Ratios," Fifth European Rotorcraft and Powered Lift Aircraft Forum, Paper No. 32, Amsterdam, the Netherlands, Sept. 1979.

11. Weidman, P. D., F. K. Browand, "Analysis of a Simple Circuit for Constant Temperature Anemometry." Journal of Physics E: Scientific Instruments. July 1975, Vol. 8, No. 7, pp. 553-560.

APPENDIX A

FOURIER SERIES AVERAGING METHOD

Early in this work many problems were encountered in attempts to reproduce the data of previous tests. A possible solution originated from the suspicion that the data reduction technique of the computer program, COSPX, was in error. Specifically, it was believed that the method of arithmetically averaging the voltage signals from each sensor over 14 revolutions using the assumption of periodicity may have been incorrect. This would be the case if the period of the signal was not a constant within reasonable limits (~ ± 5°) because a cancellation effect would result from the averaging process.

In an effort to eliminate this suspected problem, a Fourier Series averaging technique was considered. In principle this approach entails fitting a Fourier Series to the signal for each of the 14 revolutions of the rotor in order that the waves may be shifted prior to their addition and subsequent averaging. The analysis for this method proceeds as follows.

In order to obtain the Fourier coefficients, a system subroutine is used which accepts as input the data points and returns the coefficients a_n and b_n of equation A-1.

$$F(t)_i = a_o + a_n \cos(\frac{2n\pi}{T} t) + b_n \sin(\frac{2n\pi}{T} t) \qquad \text{A-1}$$

If the number of data points per revolution is N then the number n is equal to N/2 as stipulated by the subroutine. So, in this particular case n = 36.

Equation A-1 can also be expressed in the form

$$F(t)_i = f_o + C_n \sin(\frac{2\pi n}{T} t + \phi_n) \qquad \text{A-2}$$

where

$$C_n = \overline{a_n^{\,2} + b_n^{\,2}}$$

$$\phi_n = \tan^{-1} \frac{A}{B}$$

Recall that there are 14 equations in the form of A-2, one for each revolution.

Next, in order to match the peaks, an average value of ϕ_n for the first harmonic is obtained from

$$\bar{\phi}_1 = \frac{\sum_{i=1}^{14} \phi_{1i}}{14} \qquad \text{A-3}$$

where $\bar{\phi}_1$ = the average first harmonic phase shift

Once the $\bar{\phi}_1$ value is calculated, each wave can be phase shifted to $\bar{\phi}_1$ by shifting all terms by the amount $\Delta\phi_i$ given by

$$\Delta\phi_i = \bar{\phi}_1 - \phi_{1_i} \qquad i = 1,14 \qquad \text{A-4}$$

Now each wave can be reconstructed in the following manner

$$F(t)_{i_{\text{shifted}}} = C_o + \sum_{n=1}^{36} C_n \sin(\frac{2\pi n}{T} t + \phi_n + \Delta\phi_i)$$

where n = 1,36

The final step is to reduce the 14 equations to 1 by simply arithmetically averaging the values of C_n and $(\phi_n + \Delta\phi_i)$.

For example: $$\bar{C}_1 = \frac{\sum_{i=1}^{14} C_{1_i}}{14} \qquad \text{A-5}$$

Certainly all the first terms will have the same value of ϕ_n since this was dictated earlier by equations A-3 and A-4, but the remaining 35 ϕ_n values are averaged as in equation A-5.

In principle the method appears valid owing to the fact that the fundamental frequency in the signal is the prominent 2 per revolution blade frequency. Though in itself, the method did not give good results, it spurred interest in obtaining knowledge of the intermediate results in the program COSPX.FTN. This interest led to the development of a program which plots the voltage signal for each of the 14 revolutions against blade angle. Shown in Figure A-1 are the voltage traces of two revolutions from sensor A and the average obtained from all 14 revolutions. This plot shows only a small cancellation effect if any, and consequently the method was deemed adequate. This is not to say that the Fourier Series method may not provide even better results provided the believed software problem can be eliminated. One inherent problem exists in using this approach and that is the limitation of the system frequency response to the number of harmonics times the blade frequency which is ~ 1300 Hz. It was in part for this reason that the compensators mentioned in 2.2 were removed from the system.

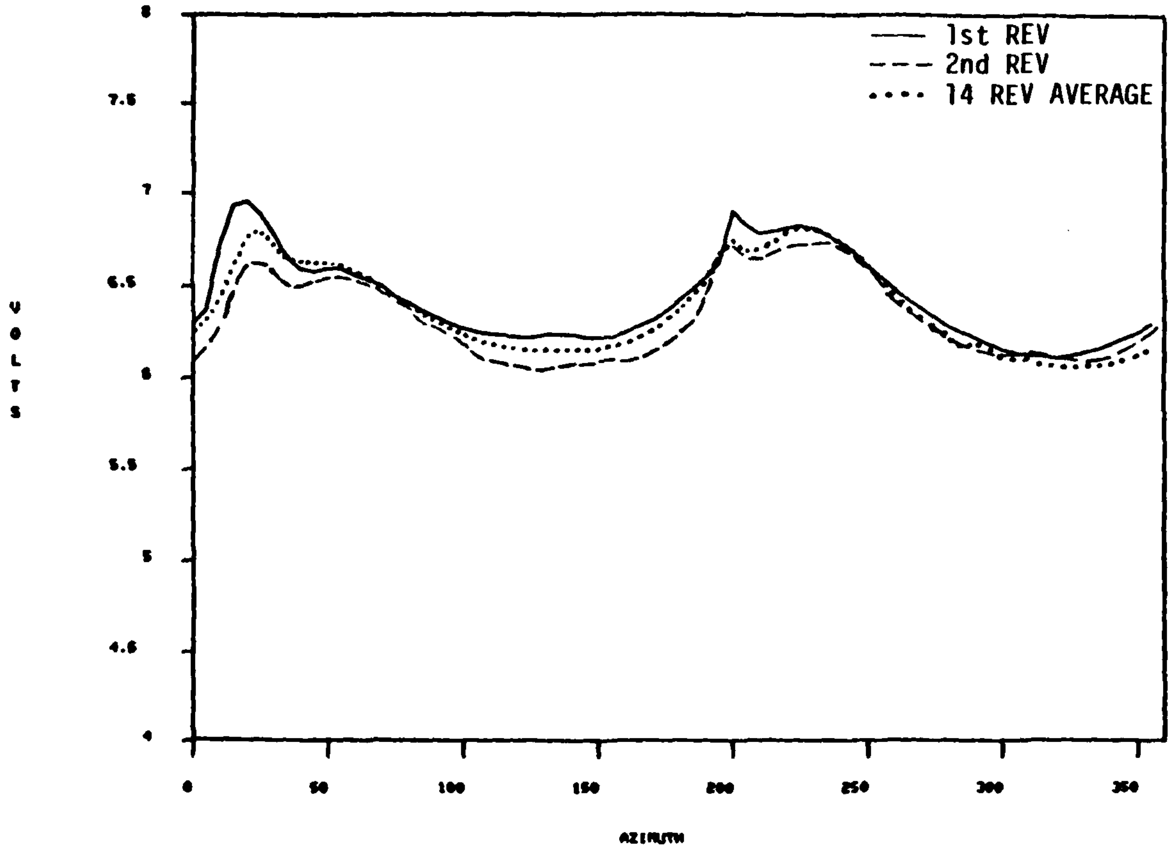


Figure A-1. Arithmetic Averaging of Blade Revolutions

APPENDIX B

REDESIGNED ANEMOMETER BRIDGE CIRCUIT

Redesign of the DISA 55D05/102C circuits was prompted by circuit transients which resulted in probe destruction on two occasions. An operational circuit for use as an alternative to these units is presented in Figure B-1. The circuit is essentially the same as the DISA circuits except that the individual transistors have been replaced by an RCA CA3140AE operational amplifier and an ARCHER VN67AF power transistor. The power transistor is necessary to supply the compensating current to the bridge.

Although this circuit is notably simplified its performance was not sufficiently improved to warrant replacement of the DISA circuits. For instance, though frequency response was about equal, the new circuit measured 2 to 5 times the noise level (light bulb tested in place of probe) when measured using the Spectral Dynamics UA500 spectrum analyzer. Due to lack of time the DISA units were repaired and perfection of the new circuit was not pursued. One suggestion for improvement of the circuit would be the substitution of a Teledyne model 1426 op-amp which has a lower noise specification than the RCA op-amp.

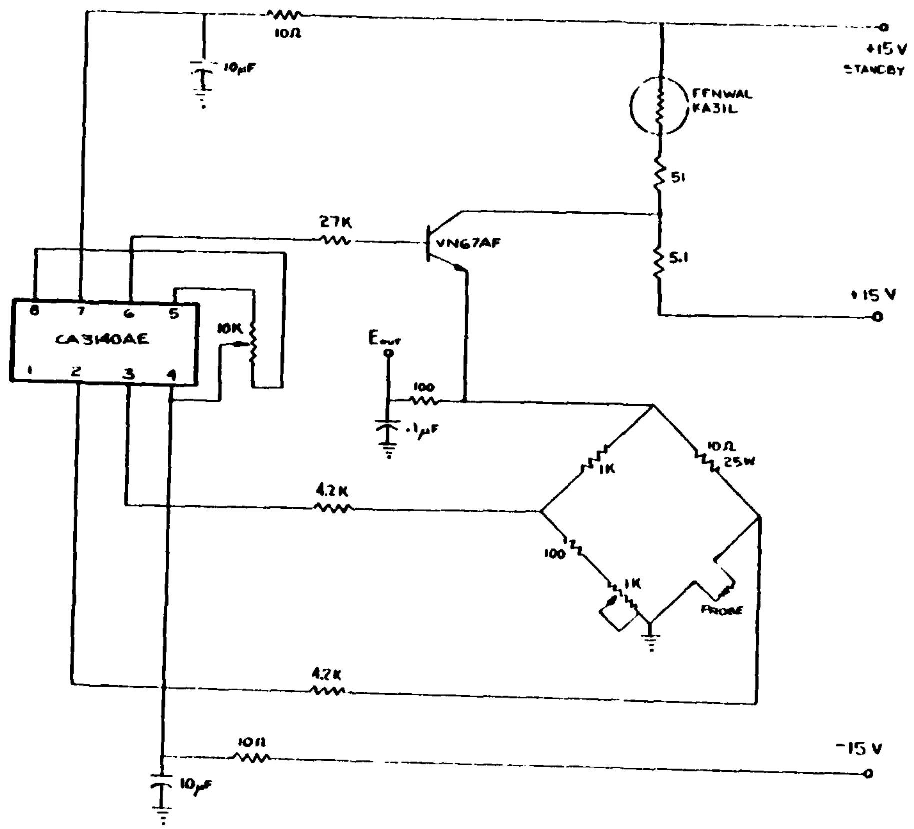


Figure B-1. Redesigned Constant Temperature Anemometer Control Circuit

APPENDIX C

COMPUTER PROGRAMS

The programs presented here deal with the various aspects of data manipulation necessary to produce the vector plots and azimuth-varying velocity plots of Figs. 4-1 to 4-7. The function of each program and important points are discussed at the beginning of each program listing. The listings are in alphabetical order.

The program EZPLOT.FTN was used to create the plots of Fig. 4.7. The tabular data for these curves, which is also provided by the program, is given in Fig. C-1.

The curves of Fig. A-1 were created by running VOLTS.FTN to plot the average of the 14 revolutions and EZVOL.FTN to plot each individual revolution.

The programs SHIFT.FTN and VECTOR.FTN were used to plot Figs. 4-1 through 4-6.

POINT 444

V_x	-12.60	-13.20	-13.20	-13.60	-14.00	-14.40
	-15.00	-15.40	-15.60	-15.80	-16.00	-16.20
	-16.40	-16.60	-16.60	-16.80	-16.80	-17.00
	-17.00	-16.80	-16.80	-16.80	-16.80	-16.80
	-16.80	-16.60	-16.60	-16.60	-16.40	-16.40
	-16.20	-16.00	-16.00	-15.80	-15.60	-15.40
	-15.40	-15.20	-15.20	-14.80	-14.60	-14.40
	-14.40	-14.40	-14.20	-13.80	-13.60	-13.60
	-13.20	-13.20	-13.00	-13.00	-12.80	-12.40
	-12.20	-12.00	-11.80	-11.40	-11.00	-10.80
	-10.80	-11.00	-11.40	-11.60	-12.00	-12.20
	-12.40	-12.20	-12.20	-12.20	-12.20	-12.40
V_y	5.80	4.60	5.00	4.40	3.80	4.00
	3.80	3.40	2.80	2.00	1.60	1.80
	2.00	1.60	1.20	1.60	1.80	1.80
	1.60	1.80	2.00	2.00	2.00	2.00
	2.00	2.20	2.20	2.20	2.00	2.00
	2.00	2.00	2.20	2.20	2.20	2.40
	2.20	2.20	2.20	2.00	2.20	2.20
	2.20	2.40	2.60	2.40	2.80	3.40
	3.80	3.20	2.80	2.20	2.00	2.20
	2.00	2.00	2.20	2.40	2.40	2.60
	2.80	2.60	2.20	2.20	2.80	3.40
	3.00	3.00	3.40	4.60	5.60	5.60
V_z	0.40	0.00	-0.40	-0.40	-0.20	-0.80
	-0.60	-0.20	0.20	0.40	0.60	0.40
	0.40	0.80	2.80	3.00	3.00	3.20
	3.40	3.60	3.80	4.00	4.20	4.40
	4.60	4.60	4.80	5.00	5.20	5.40
	5.60	5.80	5.80	6.00	6.00	6.00
	6.00	6.20	6.20	6.40	6.40	6.40
	6.60	6.40	6.20	6.80	6.40	6.00
	5.60	5.80	5.60	5.40	5.40	5.40
	5.80	5.80	5.80	5.60	5.80	6.00
	5.60	5.20	4.80	4.40	4.00	3.60
	3.60	3.40	2.80	0.00	0.40	0.40

POINT 462

V_x	-25.20	-28.20	-27.80	-27.80	-27.60	-27.80
	-27.60	-27.20	-26.60	-26.20	-26.00	-26.00
	-25.50	-25.60	-25.40	-25.40	-25.00	-24.60
	-24.90	-25.00	-24.60	-24.40	-24.20	-24.20
	-23.60	-23.20	-22.80	-22.40	-22.60	-22.80
	-22.60	-22.00	-21.80	-22.00	-22.00	-22.80
	-23.60	-24.00	-24.00	-24.20	-24.80	-25.20
	-25.20	-24.60	-25.20	-27.20	-28.40	-29.20
	-30.00	-30.80	-32.00	-33.60	-34.40	-34.80
	-35.40	-36.00	-36.40	-36.60	-36.40	-35.60
	-35.00	-34.60	-34.00	-33.20	-32.40	-31.60
	-31.00	-30.20	-29.60	-29.20	-29.00	-28.80
V_y	2.30	2.20	2.60	3.40	4.20	3.80
	3.90	4.00	3.00	3.00	3.60	4.80
	4.20	3.80	4.20	4.60	5.20	4.40
	4.00	3.60	3.80	4.40	4.20	4.20
	4.60	6.00	6.80	6.00	5.00	4.60
	4.60	4.90	5.40	5.80	5.20	5.80
	6.00	5.60	4.80	4.20	3.60	3.80
	4.00	3.40	3.20	3.80	4.40	5.00
	5.60	5.60	4.80	5.00	5.40	5.60
	5.20	4.20	4.00	4.60	4.40	3.60
	2.80	3.40	4.00	3.40	2.80	2.60
	3.20	3.20	3.20	2.20	2.40	2.80
V_z	6.20	6.60	7.20	7.00	6.80	7.00
	7.00	7.00	7.40	7.60	7.60	7.60
	7.90	8.00	8.00	8.60	9.40	10.80
	12.00	12.80	13.40	13.60	13.40	13.40
	14.00	14.20	14.60	15.20	15.60	16.00
	16.60	17.20	17.80	18.20	19.80	17.50
	17.40	17.40	18.80	20.00	20.40	20.80
	22.00	23.80	24.20	23.60	23.20	23.20
	23.00	23.00	22.60	19.80	18.20	16.30
	15.00	13.40	12.40	11.00	10.40	9.40
	8.40	6.80	5.60	5.40	5.80	5.60
	5.00	4.80	4.60	5.20	5.60	6.00

Figure C-1 Tabular Data of Fig. 4-7 (Values in ft/sec)

POINT 480

V_x	-25.80	-25.60	-25.20	-25.20	-25.20	-25.20
	-25.20	-25.20	-25.60	-25.80	-26.00	-26.20
	-25.80	-25.60	-25.80	-26.00	-26.20	-26.20
	-26.20	-26.40	-27.20	-27.20	-27.60	-28.20
	-28.40	-28.60	-28.80	-29.20	-29.60	-30.40
	-30.80	-30.60	-30.80	-31.00	-31.20	-31.40
	-31.60	-31.90	-31.60	-31.40	-31.60	-31.40
	-31.20	-30.60	-30.60	-30.00	-29.80	-29.60
	-29.00	-28.20	-27.20	-26.60	-26.60	-26.40
	-26.00	-25.60	-25.00	-25.00	-24.80	-24.40
	-24.00	-23.80	-24.00	-24.20	-25.00	-25.00
	-25.80	-25.40	-26.40	-25.40	-25.60	-25.80
V_y	3.40	3.20	3.00	3.40	3.80	4.20
	4.20	3.00	3.20	3.80	4.00	3.90
	3.80	3.60	3.60	3.80	3.60	3.20
	3.00	3.20	4.00	4.00	4.00	3.60
	3.00	3.40	3.80	3.80	3.80	3.40
	3.60	3.80	3.80	4.00	4.60	4.40
	4.20	4.80	4.80	4.80	4.80	5.60
	5.60	5.20	5.60	5.40	6.40	7.00
	7.20	7.00	6.60	6.00	5.80	5.60
	5.40	5.20	5.00	5.60	6.20	6.00
	5.80	4.20	3.40	3.00	2.60	2.40
	2.40	2.60	2.80	2.80	3.60	4.20
V_z	9.40	10.80	12.20	12.80	12.60	13.20
	14.40	15.00	14.80	14.80	15.40	16.40
	17.40	18.20	17.40	17.00	16.40	16.40
	16.60	16.40	15.60	15.80	16.00	15.60
	15.60	15.60	15.20	14.80	14.60	13.80
	13.40	13.60	13.80	13.80	13.60	13.80
	14.40	14.40	14.20	16.00	15.60	15.40
	15.60	16.00	16.00	16.40	15.40	14.80
	15.20	16.60	17.40	17.60	16.30	16.20
	15.80	15.20	15.40	14.20	13.40	12.20
	11.20	11.40	11.80	11.60	10.90	10.00
	10.20	10.60	10.20	9.80	9.40	9.00

POINT 498

V_x	-40.00	-40.20	-40.00	-39.80	-39.80	-39.80
	-39.40	-39.80	-39.60	-40.20	-40.60	-40.00
	-38.60	-40.40	-41.40	-41.40	-42.40	-41.80
	-41.60	-40.60	-40.00	-40.40	-40.80	-40.60
	-39.80	-39.20	-39.40	-39.60	-39.20	-39.00
	-37.80	-38.60	-39.00	-37.00	-37.00	-37.00
	-36.40	-36.00	-36.20	-35.60	-34.60	-34.40
	-34.00	-32.80	-35.80	-38.60	-39.00	-37.40
	-37.00	-37.20	-36.20	-36.00	-36.60	-36.60
	-36.40	-35.60	-35.80	-36.80	-37.60	-38.40
	-38.20	-36.20	-38.00	-38.40	-38.40	-38.40
	-38.60	-38.40	-36.80	-39.40	-39.80	-39.80
V_y	-0.20	-0.20	-0.20	-0.60	-0.80	-0.80
	-1.40	-1.20	0.20	-0.40	0.00	0.00
	0.20	-2.00	-1.20	-0.60	0.60	0.00
	0.60	1.20	1.20	0.60	1.60	1.60
	1.20	0.40	0.60	1.00	0.60	0.60
	0.20	1.60	1.40	1.00	1.00	0.60
	0.00	0.20	0.20	-0.20	-0.80	-1.00
	-1.00	0.60	-0.40	-4.60	-5.80	-6.20
	-5.60	-4.80	-0.60	0.20	-2.40	-1.80
	-1.60	-1.80	-1.60	-1.40	-1.20	-0.40
	-0.60	-0.20	0.00	-0.20	0.00	0.20
	0.20	0.20	0.40	0.40	0.00	-0.20
V_z	31.40	31.60	32.00	32.80	33.00	33.20
	34.00	34.20	35.60	36.60	36.60	36.20
	35.80	32.80	30.40	29.60	26.60	24.80
	23.40	22.60	22.40	21.80	19.60	19.40
	20.20	21.60	22.00	22.40	23.60	24.60
	26.60	26.00	26.40	27.80	27.80	27.60
	28.40	28.60	28.80	30.20	32.00	33.00
	33.80	36.20	33.60	31.20	32.40	34.40
	35.20	34.60	35.00	35.20	34.20	33.60
	34.00	35.00	34.60	33.60	32.20	31.00
	30.80	30.80	31.00	30.40	30.60	30.40
	30.60	31.20	31.20	31.00	31.20	31.40

Figure C-1 continued

POINT 516

	-24.60	-24.40	-24.20	-24.00	-24.00	-23.60
	-25.20	-28.00	-29.80	-31.20	-32.00	-33.20
	-34.00	-34.40	-35.20	-36.00	-36.80	-37.80
	-39.00	-40.60	-42.40	-44.60	-46.80	-48.40
	-45.50	-42.60	-42.20	-42.80	-43.60	-40.40
	-34.40	-33.20	-31.60	-30.00	-28.80	-27.80
	-26.80	-26.00	-25.60	-25.00	-24.60	-24.20
V_x	-23.80	-23.60	-23.60	-25.80	-28.40	-29.80
	-30.80	-31.60	-32.80	-33.60	-34.40	-35.20
	-36.20	-37.20	-38.20	-39.00	-41.20	-44.20
	-46.60	-47.80	-44.60	-40.60	-37.40	-34.00
	-29.00	-28.40	-27.80	-27.00	-26.20	-25.60
	-6.60	-6.40	-6.20	-5.40	-4.80	-4.40
	-4.20	-4.00	-0.40	0.00	0.40	0.60
	-2.00	-2.00	-1.80	-1.40	-1.00	-0.60
	-0.20	0.00	0.80	1.60	0.60	0.00
	-6.20	0.40	-2.60	0.00	-7.20	-9.40
V_y	-10.20	-10.40	-10.20	-9.20	-8.60	-7.80
	-7.80	-6.20	-6.60	-6.00	-5.40	-4.50
	-4.20	-4.00	-0.40	-0.20	0.00	0.40
	0.60	0.80	0.80	-1.80	-1.80	-1.60
	-1.20	-0.80	-0.20	-0.20	-0.80	-2.60
	-6.40	-7.80	-7.00	-0.80	-4.80	-5.60
	-7.00	-7.20	-7.80	-7.20	-7.40	-7.40
	39.80	39.40	39.00	38.80	38.60	38.20
	36.80	34.60	33.00	31.80	31.60	30.00
	29.20	28.80	28.20	27.60	27.20	26.80
	26.60	26.20	25.80	26.20	27.00	28.40
	32.60	37.20	41.60	45.60	48.20	51.60
V_z	55.20	53.20	50.80	48.20	46.20	44.60
	42.80	41.80	41.00	40.20	39.60	39.00
	38.40	38.20	32.00	36.20	34.20	33.00
	32.20	31.60	30.80	30.40	30.20	30.00
	29.80	29.30	29.60	30.60	30.00	28.40
	26.60	26.60	30.40	35.40	39.40	42.80
	46.60	45.80	44.80	43.60	42.20	41.00

POINT 534

	-21.40	-22.40	-23.40	-24.20	-24.00	-22.80
	-22.60	-23.20	-23.40	-23.20	-22.60	-21.60
	-20.60	-19.40	-18.20	-17.00	-15.80	-14.40
	-12.50	-10.20	-10.80	-11.40	-12.20	-12.60
	-13.20	-13.60	-14.20	-14.60	-14.60	-14.80
	-15.20	-15.20	-15.20	-17.00	-17.40	-17.40
	-18.60	-19.60	-21.00	-21.80	-21.60	-20.80
V_x	-20.60	-21.20	-21.40	-21.00	-20.40	-19.80
	-19.20	-18.40	-17.60	-16.20	-14.80	-13.00
	-10.60	-11.20	-11.80	-12.40	-13.00	-13.40
	-13.50	-14.00	-14.20	-14.40	-14.60	-14.80
	-15.00	-15.20	-16.80	-18.60	-19.60	-20.40
	-3.00	-2.60	-0.40	0.00	1.00	0.60
	0.00	0.20	0.60	0.60	0.20	0.40
	0.00	0.00	0.00	-0.40	-2.00	-2.20
	-2.60	-2.80	-3.20	-3.20	-3.40	-3.60
	-3.66	-3.40	-3.20	-3.20	-3.00	-3.00
V_y	-3.00	-2.80	-0.80	-0.80	-0.60	-0.40
	-0.60	-2.20	-0.60	0.40	0.80	0.00
	-0.40	0.00	0.00	0.20	0.00	-0.20
	-0.60	-2.40	-3.00	-3.20	-3.20	-3.40
	-3.40	-3.40	-3.40	-3.40	-3.40	-3.60
	-3.80	-3.60	-3.40	-3.40	-3.60	-3.40
	-3.20	-3.00	-3.00	-3.00	-3.60	-3.20
	21.60	21.00	20.40	19.80	18.20	16.20
	14.20	12.40	11.20	10.00	9.00	8.00
	7.40	7.40	7.60	8.40	9.60	11.20
	13.20	16.40	17.60	18.60	19.60	20.40
	21.20	22.00	22.80	23.40	23.50	24.00
V_z	24.40	24.60	24.60	23.60	23.40	23.80
	23.00	22.00	20.80	19.80	18.20	16.60
	15.00	13.40	12.20	11.60	10.80	10.20
	9.20	9.00	9.20	10.20	11.20	14.00
	17.20	18.00	19.00	20.00	20.80	21.60
	22.20	22.60	23.00	23.40	23.60	24.00
	24.40	24.60	23.60	22.80	22.40	22.00

Figure C-1 concluded

AVG.FTN

```
C       **************************************************************

C       THIS PROGRAM IS DESIGNED TO MANIPULATE THE DATA OF AN
C       EXISTING INSTANTANEOUS DATA SET WITH THE END RESULT BEING
C       AN AVERAGE DATA SET CALLED "AVCASE.DAT".  THE PROGRAM
C       FIRST DECODES THE DATA OF "CASESH.DAT" WHICH IS IN ITS
C       9 DIGIT INTEGER FORM.  IT THEN ADDS ALL THE VX,VY AND VZ
C       DATA FOR THE REVOLUTION AND DIVIDES BY THE NUMBER OF DATA
C       POINTS IN THE REVOLUTION.(CURRENTLY SET TO EVERY 10 DEGREES)
C       THE DO 40 STATEMENT CAN BE CHANGED IF THE USER WISHES SOME
C       OTHER DEGREE INCREMENT.  FOR INSTANCE: FOR EVERY 5 DEGREES
C       THE STATEMENT WOULD READ, DO 40 AZ=1,72,1.  ALSO THE DIVISORS
C       IN THE AVERAGING STATEMENTS MUST BE CHANGED.
C
C       ONCE THE AVERAGE VX,VY,VZ HAVE BEEN DETERMINED THE RESULT
C       FOR EACH POINT IS CODED ONCE AGAIN IN THE 9-DIGIT INTEGER
C       FORM IN THE DATA FILE "AVCASE.DAT".

C       IN ORDER TO USE THIS PROGRAM, IT MUST FIRST BE COMPILED
C       AND LINKED.  THE PROCEDURE FOR THIS IS AS FOLLOWS:
C
C       YOU  COM AVG
C       COMPUTER RETURNS >
C       YOU  FTB
C       COMPUTER RETURNS FTB>
C       YOU  AVG=AVG
C       COMPUTER RETURNS FTB>
C       YOU  /
C       COMPUTER RETURNS ENTER OPTIONS
C       YOU  MAXBUF=4800
C       COMPUTER RETURNS FTB>
C       YOU  //
C       COMPUTER RETURNS >
C       YOU  RUN AVG
C
C       THE PROGRAM WILL NOW RUN AND THE DO LOOP VARIABLES
C       WILL APPEAR ON THE SCREEN TO INDICATE THE STATUS.
C
C       ****************************************************


        INTEGER*4 IDATA(72),AZ,IX1,IY1,IZ1,IX2,IY2,IZ2,I9
        1 ,NDATA(1188),IPT
        OPEN(UNIT=2,NAME='ARRAY.DAT',TYPE='OLD',ACCESS='DIRECT')
        OPEN(UNIT=3,NAME='AVCASE.DAT',TYPE='NEW',ACCESS='DIRECT'
        1 ,RECORDSIZE=1)
```

```
      CALL INITT(960)

      DO 100 IPT=1,1188
      WRITE(5,1)IPT
  1   FORMAT(5X,'POINT = ',I4)
      IX2=0
      IY2=0
      IZ2=0

      DO 40 AZ=2,72,2
      READ(2'IPT)(IDATA(N),N=1,72)
      I9=IDATA(AZ)
      IX1=I9/1000000
      IY1=(I9-IX1*1000000)/1000
      IZ1=I9-IX1*1000000-IY1*1000
      IX1=IX1+IX2
      IY1=IY1+IY2
      IZ1=IZ1+IZ2
      IX2=IX1
      IY2=IY1
      IZ2=IZ1
 40   CONTINUE

C     WRITE(5,*)IX1,IY1,IZ1
C *** REMEMBER TO CHANGE THE DIVISOR IN THE FOLLOWING
C *** THREE STATEMENTS WHEN CHANGING THE NUMBER OF
C *** POINTS.

      IX1=IX2/36
      IY1=IY2/36
      IZ1=IZ2/36

C     WRITE(5,*)IX1,IY1,IZ1
      NDATA(IPT)=1000000*IX1+1000*IY1+IZ1
      WRITE(3'IPT)NDATA(IPT)
      WRITE(5,35)NDATA(IPT)
 35   FORMAT(3X,'NDATA = ',I10)
 100  CONTINUE

      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      STOP
      END
```

COSPX.FTN

```
C *****************************************************

C       THE PROGRAM COSPX.FTN IS DESIGNED TO GATHER INSTANT
C       VELOCITY DATA FOR AN ENTIRE CASE.  THE OPERATOR
C       INSTRUCTIONS ARE THOSE USED WHEN TAKING AN EN-
C       TIRE CASE.  THIS DOES NOT MEAN THAT THE PROGRAM
C       CANNOT BE USED TO TAKE DATA FOR OTHER THAN AN
C       ENTIRE CASE.  IF LESS DATA IS BEING TAKEN JUST
C       IGNORE THE INSTRUCTIONS FOR LOCATING THE PROBE.

C       THE DATA REDUCTION METHOD USED IS THE COSINE
C       TO THE P POWER METHOD.  THE PROGRAM CAN BE SET
C       TO USE EITHER CONSTANT VALUES OF P FOR EACH
C       SENSOR OR TO USE AN ITERATIVE SCHEME BETWEEN
C       U AND P IF THE CALIBRATION DATA WARRANTS THIS.

C       THE RESULT OF THE PROGRAM IS A DATA SET CALLED
C       ARRAY.DAT WHICH IS A DIRECT ACCESS FILE AND
C       THEREFORE CANNOT BE PRINTED OUT USING THE LIST
C       COMMAND.  THE STORED DATA IS IN A 9-DIGIT INTEGER
C       CODE.  WHEN DECODED(BY PROGRAMS SUCH AS EZPLOT OR
C       VECTOR)THE RESULT IS VX,VY AND VZ FOR EACH POINT
C       IN THE TUNNEL.  THERE ARE 1188 RECORDS IN THE
C       DATA SET ARRAY.DAT.  EACH RECORD HAS A RECORDSIZE
C       OF 72 SUCH THAT VX,VY AND VZ CAN BE RECORDED AT
C       5 DEGREE INCREMENTS OF THE ROTOR FOR ONE ROTOR
C       REVOLUTION.

C *******************************************************
C ***** THE FOLLOWING VARIABLES ARE DECLARED INTEGER*4
C ***** BECAUSE THEY ARE INVOLVED IN THE 9-DIGIT INTEGER
C ***** CODING SCHEME.

        INTEGER*4 ARRAY(72),NLOC,NVEL,IVX,IVY,IVZ
        1 ,I,IX,IY,IZ

C ***** THE ARRAY BUFFER IS USED TO STORE THE VOLTAGES
C ***** DELIVERED BY THE A/D CONVERTER FROM CHANNELS A
C ***** B AND C OF THE ANEMOMETER.

        INTEGER BUFFER(1024,3)

C ***** THE ARRAYS EAANG, EBANG AND ECANG ARE USED TO
C ***** STORE THE VOLTAGES FROM A B AND C FOR 14 ROTOR
C ***** REVOLUTIONS.

C ***** THE ARRAY V WILL BE USED TO STORE VEFF A, VEFF B
C ***** AND VEFF C.  THE ARRAY U WILL BE USED TO STORE THE
C ***** VALUES OF UA,UB, AND UC.
```

```
        DIMENSION EAANG(14),EBANG(14),ECANG(14),V(3),U(3)
        1 ,CSINE(3),SNE(3)
        WRITE(5,11)
11      FORMAT(3X,' IS THIS THE BEGINNING OF A CASE? ',/,
        1 ' IF YES PRINT A  Y  ,IF NOT PRINT A  N ')
        READ(5,111) ANS
111     FORMAT(1A1)
        IF(ANS.EQ.'Y') GO TO 1112
        OPEN(UNIT=1,NAME='OLDLOC.DAT',TYPE='OLD',ACCESS='DIRECT')
        OPEN(UNIT=2,NAME='ARRAY.DAT',TYPE='OLD',ACCESS='DIRECT')
        OPEN(UNIT=3,NAME='NLOC.DAT',TYPE='OLD',ACCESS='DIRECT')
        READ(1'1) I,IX,IY,IZ,IYCNTS
        GO TO 1123
C
1112    IX=28
        IY=20
        IZ=12
        I=1
        IYCNTS=0
        OPEN(UNIT=1,NAME='OLDLOC.DAT',TYPE='NEW',ACCESS='DIRECT'
        1 ,FORM='UNFORMATTED',RECORDSIZE=5)
        OPEN(UNIT=2,NAME='ARRAY.DAT',TYPE='NEW',ACCESS='DIRECT'
        1 ,FORM='UNFORMATTED',RECORDSIZE=72)
        OPEN(UNIT=3,NAME='NLOC.DAT',TYPE='NEW',ACCESS='DIRECT'
        1 ,FORM='UNFORMATTED',RECORDSIZE=1)
C
1123    IF(IY.GE.12) WRITE(5,113) IX,IY,IZ
113     FORMAT(3X,' THE PROBE SHOULD BE NOW ON HOLDER  A',/,
        1 ' AT LOCATION:  ',//,' X= ',I3,/,' Z= ',I3,/,' Y= ',I3)
C
        IF(IY.LE.8.AND.IY.GE.4) WRITE(5,114) IX,IY,IZ
114     FORMAT(3X,' THE PROBE SHOULD BE NOW ON HOLDER B',/,
        1 'AT LOCATION:',//,' X= ',I3,/,' Y= ',I3,/,' Z= ',I3)
C
        IF(IY.LE.0.AND.IY.GE.-8) WRITE(5,115) IX,IY,IZ
115     FORMAT(3X,' THE PROBE SHOULD BE NOW ON HOLDER C',/,
        1 ' AT LOCATION:',//,' X= ',I3,/,' Y= ',I3,/,' Z= ',I3)
C
        IF(IY.LE.-12) WRITE(5,116) IX,IY,IZ
116     FORMAT(3X,' THE PROBE SHOULD BE NOW ON HOLDER D',/,
        1 ' AT LOCATION:  ',//,' X= ',I3,/,' Y= ',I3,/,' Z= ',I3)
        WRITE(5,650)
650     FORMAT(3X,' WHEN IT IS THERE PRINT OK')
        READ(5,651) LOC
        IF(LOC.NE.'OK') GO TO 1123
651     FORMAT(1A2)
```

```
C
         MX=(28-IX)/4+1
         MY=(20-IY)/4+1
         MZ=(12-IZ)/3+1
C ***** THE SUBROUTINE SETAD1 IS USED TO SET THE PARAMETERS
C ***** OF THE A/D CONVERTER.

         CALL SETAD1(3,3,1,5)
         DO 101   JY=MY,11
         DO 100   JZ=MZ,6
         DO 99    JX=MX,18
C
        .IND=1
         IF(ABS(IX).LE.12.AND.IZ.LE.3.AND.ABS(IY).LE.12)

C ***** THE SUBROUTINE ZERO SETS ALL POINTS IN THE ROTOR
C ***** INTERFERENCE VOLUME EQUAL TO ZERO.

         1 CALL ZERO(ARRAY,IND)
         IF(IND.EQ.0) GO TO 91
652      WRITE(5,117)
117      FORMAT(3X,' NOW PUSH THE SWITCH ON INTERFACE ')

C ***** THE SUBROUTINE GETAD IS USED TO ACTUATE THE A/D
C ***** CONVERTER TO FILL THE ARRAY, BUFFER.

         CALL GETAD(BUFFER)
         WRITE(5,5000)
5000     FORMAT(3X,' BUFFER WAS FILLED')
         DO 85 J=1,72
C           (72=360/5)
         DO 9 M=1,3
         IA=J
         DO 8 JJ=1,14
C           14-REVOLUTIONS
         IF(M.EQ.1) EAANG(JJ)=FLOAT(BUFFER(IA,1))/(32768./8.)
         IF(M.EQ.2) EBANG(JJ)=FLOAT(BUFFER(IA,2))/(32768./8.)
         IF(M.EQ.3) ECANG(JJ)=FLOAT(BUFFER(IA,3))/(32768./8.)
         IA=IA+72
8        CONTINUE

C ***** THE SUBROUTINE AVERAG IS USED TO GET AVERAGE
C ***** VOLTAGES FOR ONE REVOLUTION FROM THE 14 REVS
C ***** STORED PREVIOUSLY.

         IF(M.EQ.1) CALL AVERAG(EAANG,AVA)
         IF(M.EQ.2) CALL AVERAG(EBANG,AVB)
         IF(M.EQ.3) CALL AVERAG(ECANG,AVC)
9        CONTINUE
C        VOLTAGE TO VELOCITY CALCULATION
C
```

```
C        USE OF CALIBRATION CURVES TO OBTAIN THE CORRESPONDING
C        EFFECTIVE VELOCITIES
C
C        CALIBRATION CONSTANTS FOR PARTICULAR PROBE
C
         A0=.7552
         A1=1.7513
         A2=.7304
         A3=.01033
         A4=.001053
         V(1)=A0+A1*AVA+A2*AVA**2+A3*AVA**3+A4*AVA**4
C
         B0=.1367
         B1=1.833
         B2=.8454
         B3=-.02726
         B4=.004034
         V(2)=B0+B1*AVB+B2*AVB**2+B3*AVB**3+B4*AVB**4
C
         C0=.6043
         C1=1.8375
         C2=.7809
         C3=-.01365
         C4=.003839
         V(3)=C0+C1*AVC+C2*AVC**2+C3*AVC**3+C4*AVC**4
C
C WRITE STATEMENTS TO PRINT VOLTAGES AND UA UB UC
C ***** NOTICE THAT THESE STATEMENTS HAVE A "D" IN COLUMN
C ***** 1. THIS MEANS THAT IF WHEN COMPILING THE USER TYPES
C ***** " COM COSPX /DE " (DON'T INCLUDE THE QUOTES) THE "D"
C ***** STATEMENTS WILL BE INCLUDED IN THE PROGRAM. IF /DE
C ***** IS NOT USED THE PROGRAM TREATS D STATEMENTS AS
C ***** COMMENT STATEMENTS.

D        IF(J.GT.71)WRITE(6,288)AVA,AVB,AVC
 288     FORMAT(2X,'AVA = ',F7.3,' AVB = ',F7.3,' AVC = ',F7.3)
D        IF(J.GT.71)WRITE(6,289)V(1),V(2),V(3)
D 289    FORMAT(3X,'V1 = ',F7.3,' V2 = ',F7.3,' V3 = ',F7.3)
         IF(J.GT.71)WRITE(5,290)AVA,AVB,AVC
 290     FORMAT(2X,'VOLTAGES - ',3F7.3)
         IF(J.GT.71)WRITE(5,291)V(1),V(2),V(3)
 291     FORMAT(4X,'V1 = ',F7.3,' V2 = ',F7.3,' V3 = ',F7.3)
C
C
C        CALCULATE MAGNITUDE OF THE ABSOLUTE VELOCITY
C ***** NOTICE THAT THE CALL STATEMENT IS CURRENTLY COMMENTED
C ***** OUT OF THE PROGRAM BUT IF AN ITERATION BETWEEN U AND
C ***** P WERE NECESSARY THE STATEMENT CAN BE EASILY CHANGED.
```

```
C       CALL UPITER(V,VEL,P,J)
        P=.78
C ***** VEL IS THE STREAM VECTOR  -U-
        VEL=((V(1)**(2./P)+V(2)**(2./P)+V(3)**(2./P
        1))/2.)**(P/2.)
D       IF(J.GT.71)WRITE(6,287)VEL
        IF(J.GT.71)WRITE(5,287)VEL
 287    FORMAT(3X,'VEL = ',F7.3)
C
C       FIND ANGLES IN PROBE SYSTEM OF COORDINATES
C ***** THE VARIABLE K REPRESENTS A B AND C.

        DO 15 K=1,3
        IF(K.EQ.1)P=.76
        IF(K.EQ.2)P=.80
        IF(K.EQ.3)P=.78
D       IF(J.GT.71)WRITE(6,296)P
 296    FORMAT(3X,' P = ',F5.3)
        CSINE(K)=(V(K)/VEL)**(1./P)
        IF(CSINE(K).GT.1.0) CSINE(K)=1.0
        IF(CSINE(K).GT.0.99.AND.J.GT.71)WRITE(5,297)
  297   FORMAT(3X,'COSINE IS GREATER THAN .99')
        ACSN=ACOS(CSINE(K))
        SNE(K)=SIN(ACSN)
C       IF(J.GT.71)WRITE(5,295)SNE(K)
D       IF(J.GT.71)WRITE(6,295)SNE(K)
  295   FORMAT(3X,'SINE = ',F6.3)
D       IF(J.GT.71)WRITE(6,286)CSINE(K)
 286    FORMAT(5X,'CSINE(K) = ',F6.3)
C
C       FIND PERPENDICULAR COMPONENTS OF VELOCITY
C
C       IN PROBE SYSTEM OF COORDINATES
15      U(K)=VEL*SNE(K)
        IF(J.GT.71)WRITE(6,292)U(1),U(2),U(3)
  292   FORMAT(3X,'U1 = ',F7.3,' U2 = ',F7.3,' U3 = ',F7.3)
D       IF(J.GT.71)WRITE(5,293)U(1),U(2),U(3)
D 293   FORMAT(3X,'U1 = ',F7.3,' U2 = ',F7.3,' U3 = ',F7.3)
C
C       IN TUNNEL SYSTEM OF COORDINATES
        VZ=0.57735*(U(1)+U(2)-U(3))
        VY=.7071*(U(1)-U(2))
        VX=-0.40825*(U(1)+U(2))-0.8165*U(3)
```

```
C
C CORRECTION FACTORS INCORPORATED IN VY AND VZ
C AS FUNCTIONS OF VEL.
               IF(ABS(VY).LT.1.0)GO TO 299
               VYCOR=VEL*(-.091)+.58
               VY=VY+VYCOR
  299          IF(ABS(VZ).LT.1.0)GO TO 300
               VZCOR=VEL*(.108)-.06
               VZ=VZ+VZCOR
C *****        THE FOLLOWING STATEMENTS ARE THE 9-DIGIT INTEGER
C *****        CODING SCHEME USED TO CONSERVE MEMORY SPACE.
  300          IVX=IFIX(VX*5.+0.5)+499
               IVY=IFIX(VY*5.+0.5)+499
               IVZ=IFIX(VZ*5.+0.5)+499
               NVEL=IVX*1000000+IVY*1000+IVZ
               ARRAY(J)=NVEL
D              IF(J.GT.60)WRITE(6,69)VX,VY,VZ
 69            FORMAT(3X,'VX=',1X,F7.2,2X,'VY=',1X,F7.2,2X,
               1 'VZ=',1X,F7.2)
               IF(J.GT.71)WRITE(5,68)VX,VY,VZ
 68            FORMAT(3X,'VX =',F7.2,2X,'VY =',F7.2,2X,'VZ =',F7.2)
85             CONTINUE
91             NLOC=(IX+499)*1000000+(IY+499)*1000+IZ+499
C
               WRITE(2'I) (ARRAY(K),K=1,72)
               WRITE(3'I) NLOC
               IF(IND.EQ.0) GO TO 990
               WRITE(5,991)
991            FORMAT(3X,' WAS THE PROBE IN THE CORRECT LOCATION?',//,
               1  'IF YES PUSH RETURN .  IF NOT, ENTER -N- ')
               READ(5,111) BACK
               IF(BACK.EQ.'N') GO TO 652
990            I=I+1
               IX=IX-4
               IF(IX.EQ.12.AND.IZ.LE.3.AND.ABS(IY).LE.12) IND=0
               IF(IND.EQ.0) WRITE(5,86)
86             FORMAT(3X,' YOU ARE NOW IN THE DANGEROUS AREA',/,
               1 ' MOVE THE PROBE TO Z=6 (78 CNTS),AND THEN TO X=-16')
               IF(IX.LT.-40) IX=28
               IF(IND.EQ.1) WRITE(5,96) IX
96             FORMAT(3X,' MOVE PROBE TO LOCATION',//,
               1  ' X=',I4)
               IF(IX.EQ.28.AND.IZ.LE.3.AND.ABS(IY).LE.12) WRITE(5,106)
106            FORMAT(3X,' BEFORE YOU MOVE THE PROBE TO X=28',/,
               1 ' TAKE IT HORIZONTALY TO Z=6 (78 CNTS)')
               WRITE(5,92)
92             FORMAT(3X,' IF YOU WISH TO CONTINUE PUSH RETURN',/,
               1 '    IF NOT PRINT AN --N--.  ')
               READ(5,111) ANS
               IF(ANS.EQ.'N') GO TO 1000
99             CONTINUE
```

```
          MX=1
          IZ=IZ-3
          IF(IZ.LT.-3) IZ=12
          IZCNTS=(12-IZ)*13
          WRITE(5,124) IZ,IZCNTS
124       FORMAT(3X,' MOVE PROBE HORIZONTALY TO',//,
          1 ' Z= ',I4,' EQUIVALENT TO',I4,' (CNTS)')
125       WRITE(5,126)
126       FORMAT(3X,' PUSH RETURN TO CONTINUE')
          READ(5,111) ZANS
          IF(ZANS.EQ.'N') GO TO 125
100       CONTINUE
          MZ=1
          IY=IY-4
          IYCNTS=IYCNTS+52
          IF(IY.EQ.8.OR.IY.EQ.0.OR.IY.EQ.-12) IYCNTS=0
350       FORMAT(3X,' MOVE THE PROBE VERTICALLY TO  ',/,
          1 ' Y= ',I4,' EQUIVALENT TO',I4,' (CNTS)')
          IF(IY.EQ.8) WRITE(5,360)
360       FORMAT(3X,' CHANGE PROBE TO HOLDER --B--')
          IF(IY.EQ.0) WRITE(5,370)
370       FORMAT(3X,' CHANGE PROBE TO HOLDER --C--')
          IF(IY.EQ.-12) WRITE(5,380)
380       FORMAT(3X,' CHANGE PROBE TO HOLDER --D--')
          WRITE(5,350) IY,IYCNTS
390       WRITE(5,400)
400       FORMAT(3X,' PRINT RETURN TO CONTINUE ',/,
          1 ' YOU CAN NOT TERMINATE THE SESSION FROM HERE')
          READ(5,111) YANS
          IF(YANS.EQ.'N') GO TO 390
101       CONTINUE
1000      IZCHEK=IZ-3
          IXCHEK=IX-4
          IF(IXCHEK.LT.-40.AND.IZCHEK.LT.-3)IZ=12
          IF(IXCHEK.LT.-40.AND.IZCHEK.LT.-3)IY=IY-4
          WRITE(1'1) I,IX,IY,IZ,IYCNTS
          DO 1001 IU=1,3
1001      CLOSE(UNIT=IU)
          CLOSE(UNIT=6)
          STOP
          END
C         SUBROUTINE  ITERATES BETWEEN U & P TO GET VEL=U
C
C         SUBROUTINE UPITER(V,VEL,P,J)
```

```
C
C      DIMENSION V(3),VELI(16)
C      FIRST GUESS FOR P IS 0. 6
C      P=. 6
C      VEL=((V(1)**(2. /P)+V(2)**(2. /P)+V(3)**(2. /P))/2.
C      1    )**(P/2. )
C      VELI(1)=VEL
C      DO 77 I=1,15
C      RE=(VEL-. 346)/. 885
C      P=. 593+. 0042*RE
C      VELI(I+1)=((V(1)**(2. /P)+V(2)**(2. /P)+V(3)**
C      1            (2. /P))/2. )**(P/2. )
C      IF(ABS(VELI(I+1)-VELI(I)). GT. 0. 01. AND. I. EQ. 15. AND.
C      1  (J. GT. 71))WRITE(5,283)
C 283  FORMAT(3X,'*** P HAS NOT CONVERGED')
C      IF(ABS(VELI(I+1)-VELI(I)). LT. 0. 01) GO TO 78
C 77   CONTINUE
C 78   VEL=VELI(I)
C      IF(J. GT. 71)WRITE(5,410)P
C 410  FORMAT(3X,'P =',F7. 4)
C      RETURN
C      END
C ***** NOTE THE FOLLOWING FUNCTIONS ARE USED IN THE K**2
C ***** METHOD AND HAVE BEEN COMMENTED OUT.
C      FUNCTIONS FOR K=F(U) FOR EACH SENSOR
C
C      SENSOR A
       FUNCTION UK1(U)
C
       AU0=. 2796
       AU1=-. 03397
       AU2=2. 387E-3
       AU3=-6. 541E-5
       AU4=5. 959E-7
       UK1=AU4*U**4+AU3*U**3+AU2*U**2+AU1*U+AU0
       RETURN
       END
C      SENSOR B
C
       FUNCTION UK2(U)
       BU0=. 06048
       BU1=-7. 794E-4
       BU2=5. 249E-4
       BU3=-2. 231E-5
       BU4=2. 441E-7
       UK2=BU4*U**4+BU3*U**3+BU2*U**2+BU1*U+BU0
       RETURN
       END
```

```
C       SENSOR C
C
        FUNCTION UK3(U)
        CU0=-.2898
        CU1=.06627
        CU2=-3.443E-3
        CU3=7.156E-5
        CU4=-5.227E-7
        UK3=CU4*U**4+CU3*U**3+CU2*U**2+CU1*U+CU0
        RETURN
        END
C
C
        SUBROUTINE AVERAG(A,AV)
        DIMENSION A(14),SUM(14)
        SUM(1)=A(1)
        DO 9 I=1,13
 9      SUM(I+1)=SUM(I)+A(I+1)
        AV=SUM(14)/14.
        RETURN
        END
C
C       SUBROUTINE TO SET VX=VY=VZ=0
C       WHEN THE PROBE IS IN THE DANGEROUS REGION
C
        SUBROUTINE ZERO(ARRAY,IND)
        INTEGER*4 ARRAY(72),NVEL,IVX,IVY,IVZ
        DO 10 I=1,72
        IVX=499
        IVY=499
        IVZ=499
        NVEL=IVX*1000000+IVY*1000+IVZ
10      ARRAY(I)=NVEL
        IND=0
        RETURN
        END
```

EZPLOT.FTN

```
C ************************************************************

C THE PROGRAM EZPLOT.FTN IS DESIGNED TO PLOT THE AZIMUTH
C VARYING VELOCITIES FOR X, Y AND Z.  THE RESULT OF THE
C PROGRAM IS NOT A PLOT BUT A FILE IN THE DIRECTORY CALLED
C FOR001.DAT WHICH WILL CONTAIN VX,VY OR VZ DEPENDING ON
C WHICH YOU SPECIFY IN THIS PROGRAM WHEN YOU RUN IT.

C THE PROGRAM WILL ASK YOU FOR A PARTICULAR POINT NUMBER
C WHICH MUST BE FROM 1 TO 1188.  IF YOU NEED HELP IN FINDING
C THE POINT NUMBER OF A TUNNEL COORDINATE, SEE REFTAB.FTN.

C IN ORDER TO GRAPH THE FILE FOR001.DAT YOU MUST FOLLOW THIS
C THIS SIMPLE PROCEDURE TO USE EASY GRAPHING.

C    1) AFTER THE PROGRAM HAS BEEN RUN TYPE THE FOLLOWING
C       "COPY FOR001.DAT PT.DAT"  (OMIT THE QUOTES)

C    2) AFTER THE PROMPT HAS BEEN RETURNED TYPE IN
C       "EZG"

C    3) THE SCREEN WILL BE CLEARED AND A BELL SOUNDED THEN TYPE
C       "RUN VEL"

C    4) IF YOU WANT TO CHANGE THE SCALES OR ADD TITLES SEE
C       THE MANUAL ON EASY GRAPHING

C ************************************************************

        INTEGER*4 ARRAY(72),CODVEL,IVX,IVY,IVZ
        DIMENSION ANGL(72,3)
        OPEN(UNIT=2,NAME='ARRAY.DAT',TYPE='OLD',ACCESS='DIRECT')
1       CALL INITT(960)
        WRITE(5,2)
2       FORMAT(3X,' PLEASE ENTER "I" A DATA POINT')
        READ(5,*) I
C *** THE FOLLOWING IS A DIRECT ACCESS READ STATEMENT
C *** TO READ THE DATA FROM ARRAY OF UNIT 2 CORRES-
C *** PONDING TO THE POINT NUMBER THE USER HAS SPECIFIED.
        READ(2'I) (ARRAY(J),J=1,72)
```

```
       DO 10 II=1, 72
       IF(II. LT. 2)GO TO 201
       GO TO 202
 201   WRITE(5, 99)
 99    FORMAT(3X, 'DO YOU WISH TO INCLUDE A TUNNEL CORRECTION
       1 FACTOR ? ', //3X, 'TYPE "Y" FOR YES OR "N" FOR NO')
       READ(5, 199)REPLY
199    FORMAT(1A1)
       IF(REPLY. EQ. 'N')GO TO 202
       WRITE(5, 103)
 103   FORMAT(3X, 'TYPE IN Y-CORRECTION')
       READ(5, 110)YCOR
 110   FORMAT(F3. 1)
       WRITE(5, 104)
 104   FORMAT(3X, 'TYPE IN Z-CORRECTION')
       READ(5, 120)ZCOR
 120   FORMAT(F3. 1)
 202   CODVEL=ARRAY(II)
C *** THE FOLLOWING STATEMENTS ARE THE DECODING ROUTINE
C *** FOR THE 9-DIGIT INTEGERS.

       VX=(CODVEL/1000000-499)/5
       IVX=(CODVEL/1000000)*1000000
       VY=(FLOAT((CODVEL-IVX)/1000)-499. )/5.
       IVY=((CODVEL-IVX)/1000)*1000
       VZ=FLOAT(CODVEL-IVX-IVY-499)/5.

8      ANGL(II, 1)=VX
       VY=VY+YCOR
       VZ=VZ+ZCOR
       ANGL(II, 2)=VY
       ANGL(II, 3)=VZ
10     CONTINUE
       WRITE(5, 55)(ANGL(M, 1), M=1, 72, 1)
55     FORMAT(3X, 6F9. 2)
       WRITE(5, 56)
  56   FORMAT(1X, /)
       WRITE(5, 57)(ANGL(M, 2), M=1, 72, 1)
  57   FORMAT(3X, 6F9. 2)
       WRITE(5, 58)
  58   FORMAT(1X, /)
       WRITE(5, 59)(ANGL(M, 3), M=1, 72, 1)
  59   FORMAT(3X, 6F9. 2)
       WRITE(5, 400)
400    FORMAT(3X, 'IF YOU WANT TO PLOT VX, VY OR VZ PUSH RETURN'
       1 , /, '            IF NOT TYPE "N" ')
       READ(5, 401)KEY
401    FORMAT(1A1)
```

```
      IF(KEY.EQ.'N')GOTO 1000
      WRITE(5,500)
500   FORMAT(3X,'TYPE IN VX,VY, OR VZ')
      READ(5,501)ANS
501   FORMAT(1A2)
      IF(ANS.EQ.'VX') J=1
      IF(ANS.EQ.'VY') J=2
      IF(ANS.EQ.'VZ') J=3
      IF(ANS.EQ.'VX') GOTO 777
      IF(ANS.EQ.'VY') GOTO 888
      DO 100 I=1,72
      YZ=ANGL(I,J)
      WRITE(1,*)YZ
100   CONTINUE

      GOTO 999

777   DO 101 I1=1,72
      IX=((I1-1)*5)*2
      YX=ANGL(I1,J)
      WRITE(1,*)YX
101   CONTINUE

      GOTO 999

888   DO 102 I2=1,72
      YY=ANGL(I2,J)
      WRITE(1,*)YY
102   CONTINUE
999   CONTINUE

1000  WRITE(5,901)
901   FORMAT(3X,'WOULD YOU LIKE TO PLOT A DIFFERENT POINT?',/3X,'
      1 TYPE "Y" FOR YES AND "N" FOR NO' )
      READ(5,902)LET
902   FORMAT(1A1)
      IF(LET.EQ.'Y')  GOTO 1
      CLOSE(UNIT=2)
      STOP
      END
```

EZVOL.FTN

```
C *****************************************

C THE PROGRAM EZVOL.FTN CREATES A FILE IN
C THE DIRECTORY CALLED FOR003.DAT WHICH
C CONTAINS THE VOLTAGES FOR EACH REV OF
C EITHER CHANNEL A B OR C.  PLOTTING CAN
C BE DONE USING THE EASY GRAPHING ROUTINE.

C THE PROGRAM WHEN RUN WILL ASK IF YOU WANT
C THE VOLTAGES FROM A, B OR C, AND OF WHICH
C REVOLUTION.  IF YOU WANT TO PLOT THE AVER-
C AGE OF ALL THE REVS YOU MUST RUN THE PRO-
C GRAM VOLTS.FTN.

C BEFORE YOU CAN PLOT THESE VOLTAGES YOU MUST
C RUN THE PROGRAM MAKEBUF.FTN TO CREATE THE
C DIRECT ACCESS FILE BUFFER.DAT.

C ******************************************

        INTEGER BUFFER(1024,3)
        DIMENSION FNT(72,14,3)
        OPEN(UNIT=1,NAME='BUFFER.DAT',TYPE='OLD',ACCESS='
        1 DIRECT')
        READ(1'1)  (BUFFER(N,1),N=1,1024)
        READ(1'2)  (BUFFER(N,2),N=1,1024)
        READ(1'3)  (BUFFER(N,3),N=1,1024)

        DO 5 M=1,3
        K=0
        DO 10 J=1,14
        DO 20 I=1,72
        L=K+I

C *** DECODING THE BINARY VOLTAGE VALUE.

        FNT(I,J,M)=FLOAT(BUFFER(L,M))/(32768./8.)
20      CONTINUE
        K=K+72
10      CONTINUE
 5      CONTINUE

C *** INITT(960) CLEARS THE PAGE AND SETS BAUD

        CALL INITT(960)
        WRITE(5,500)
500     FORMAT(4X,'DO YOU WANT TO PLOT VOLTAGES A,B OR C?')
        READ(5,501)REPLY
501     FORMAT(1A1)

        IF(REPLY.EQ.'A') GOTO 100
        IF(REPLY.EQ.'B') GOTO 200
        IF(REPLY.EQ.'C') GOTO 300
```

```
C  THIS IS THE STORAGE ROUTINE FOR CHANNEL -C-

300     WRITE(5,502)
502     FORMAT(4X,'WHICH REVOLUTION WOULD YOU LIKE TO PLOT,
        1  1,2,3...14 ?')
        READ(5,*)IREV
        JC=IREV

        DO 301 IC=1,72
        YC=(FNT(IC,JC,3))
        WRITE(3,*)YC
301     CONTINUE
        GOTO 700

C  THIS IS THE STORAGE ROUTINE FOR CHANNEL -B-

200     WRITE(5,511)
511     FORMAT(4X,'WHICH REVOLUTION WOULD YOU LIKE TO PLOT,
        1  1,2,...14?')
        READ(5,*)IREVB
        JB=IREVB

        DO 201 IB=1,72
        YB=FNT(IB,JB,2)
        WRITE(3,*)YB
201     CONTINUE
        GOTO 700

C  THIS IS THE STORAGE ROUTINE FOR CHANNEL -A-

100     WRITE(5,512)
512     FORMAT(4X,'WHICH REVOLUTION WOULD YOU LIKE TO PLOT
        1 ,  1,2...14?')
        READ(5,*)IREVA
        JA=IREVA

        DO 101 IA=1,72
        YA=FNT(IA,JA,1)
        WRITE(3,*)YA
101     CONTINUE

700     CLOSE(UNIT=1)
        STOP
        END
```

LOOK.FTN

```
C ************************************************
C THE PROGRAM LOOK.FTN ALLOWS THE USER TO INSPECT THE
C 9-DIGIT INTEGER DATA STORED IN THE FILE CASESH.DAT.
C THIS WOULD PRIMARILY BE USED IN A DIAGNOSTIC SENSE.

C THE FILE CASESH.DAT IS THE RESULT OF THE RUNNING O
C SHIFT.FTN.  SINCE THIS REPRESENTS AN ENORMOUS AMOUNT
C OF DATA ONLY ONE AZIMUTH POSITION IS ASKED FOR AT
C ONE TIME.  THE DATA FOR THIS AZIMUTH WILL BE STORED
C IN THE FILE FOR006.DAT.

C ***************************************************

      INTEGER*4 IDATA(18,6,11),ARRAY(114),AZ
      OPEN(UNIT=1,NAME='CASESH.DAT',TYPE='OLD',ACCESS='DIRECT')
      WRITE(5,100)
  100 FORMAT(3X,'TYPE IN THE AZIMUTH OF THE DATA(IE:1 TO 72)')
      READ(5,*)AZ
      READ(1'AZ)(((IDATA(K,I,J),K=1,18),I=1,6),J=1,11)
      WRITE(6,200)IDATA
  200 FORMAT(3X,9I10)
      CLOSE(UNIT=1)
      STOP
      END
```

MAKEBUF.FTN

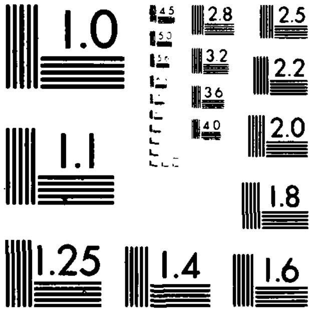
```
C *******************************************

C THE PROGRAM MAKEBUF.FTN IS DESIGNED TO FILL
C THE DIRECT ACCESS FILE BUFFER.DAT WITH THE
C VOLTAGES FROM A, B, AND C FOR ALL 14 REVS.

C THE PROGRAM MUST BE USED WITH A TRIGGER TO
C ACTIVATE THE A/D CONVERTER.  THE WIND TUNNEL
C INTERFACE UNIT OR A WAVETEK MAY BE USED.

C TO SUBSEQUENTLY PLOT THE DATA GATHERED HERE,
C RUN EITHER VOOLTS.FTN FOR THE AVERAGE OR
C EZVOL.FTN FOR EACH INDIVIDUAL REV.

C REMEMBER TO USE THE FTB LINK PROCEDURE AND
C USE A MAXBUF=4752.

C *******************************************

       INTEGER BUFFER(1024,3)
       OPEN(UNIT=1,NAME='BUFFER.DAT',TYPE='NEW',ACCESS='DIRECT'
       1 ,FORM='UNFORMATTED',RECORDSIZE=1024)

C *** SETAD1 READIES THE A/D CONVERTER

       CALL SETAD1(3,3,1,5)
       WRITE(5,200)
 200   FORMAT(3X, 'NOW PUSH THE SWITCH ON THE INTERFACE')

C *** GETAD ACTUALLY FILLS BUFFER

       CALL GETAD(BUFFER)
       WRITE(5,300)
 300   FORMAT(3X, 'BUFFER HAS BEEN FILLED')

C *** BUFFER IS BEING WRITTEN INTO UNIT ONE
       WRITE(1'1) (BUFFER(N,1),N=1,1024)
       WRITE(1'2) (BUFFER(N,2),N=1,1024)
       WRITE(1'3) (BUFFER(N,3),N=1,1024)
       READ(1'1) (BUFFER(N,1),N=1,1024)
       READ(1'2) (BUFFER(N,2),N=1,1024)
       READ(1'3) (BUFFER(N,3),N=1,1024)

C *** THE WRITE STATEMENT PRINTS EVERY 100TH VALUE
C *** OF BUFFER ON THE SCREEN.
       WRITE(5,400)((BUFFER(M,MN),M=1,1024,100),MN=1,3)
400    FORMAT(/1X,5I10,/,5I10)
       CLOSE(UNIT=1)
       STOP
       END
```

AD-A108 107 OHIO STATE UNIV RESEARCH FOUNDATION COLUMBUS F/G 1/3
DEVELOPMENT OF A COMPUTER BASED PRESENTATION OF NON-STEADY HELI--ETC(U)
SEP 81 H R VELKOFF, R R NAVARRO DAAG29-79-C-0074

UNCLASSIFIED ARO-14142.6-EX NL

2 OF 2
AD
A108107

END
DATE
FILMED
1-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REFTAB.FTN

```
C ************************************************

C THE PROGRAM REFTAB.FTN CREATES A TABLE IN THE
C DIRECTORY UNDER THE NAME FOR002.DAT.  THIS TABLE
C CONTAINS THE X,Y AND Z COORDINATES(IN INCHES) OF
C EACH OF THE 1188 POINTS IN A CASE.  THIS PROGRAM
C CAN BE LINKED BY USING "LINK REFTAB" SINCE IT IS NOT
C USING THE DIRECT ACCESS MODE.  THE FTN FILE MUST
C ALSO BE COMPILED USING "COM REFTAB" BEFORE IT CAN
C BE RUN.  TO RUN THE PROGRAM JUST TYPE "RUN REFTAB"
C IF YOU HAVE ALREADY COMPILED AND LINKED THE PROGRAM.

C ***************************************************

      DIMENSION LIST(1188,4)
      I=0
      DO 80 J=1,11
      DO 70 K=1,6
      DO 60 L=1,18
      I=I+1
      DO 50 M=1,4
      WRITE(5,*)I,J,K,L,M
      IF(M.EQ.1)LIST(I,M)=I
      IF(M.EQ.2)LIST(I,M)=28-(L-1)*4
      IF(M.EQ.3)LIST(I,M)=20-(J-1)*4
      IF(M.EQ.4)LIST(I,M)=12-(K-1)*3
 50   CONTINUE
 60   CONTINUE
 70   CONTINUE
 80   CONTINUE

      WRITE(2,90)((LIST(I,M),M=1,4),I=1,1188)
 90   FORMAT(10X,4I10)
      STOP
      END
```

REWRIT.FTN

```
C***************************************
C
C     THIS PROGRAM ALLOWS DATA TO BE WRITTEN OVER
C     POINTS THAT HAVE ALREADY BEEN TAKEN.  THE
C     USER MUST SPECIFY THE POINT, I, THE POSITION
C     IX,IY,IZ AND THE VERTICAL COUNTER READING,
C     IYCNTS.  THEN INSTVEL CAN BE RUN AS USUAL
C     USING THE "OLD CASE" ANSWER.
C
C*************************************************
C
C *** THE VALUES IX,IY,IZ ARE IN INCHES

        INTEGER*4 I,IX,IY,IZ
        OPEN(UNIT=1,NAME='OLDLOC.DAT',TYPE='NEW',ACCESS='DIRECT'
        1 ,FORM='UNFORMATTED',RECORDSIZE=5)
        WRITE(5,10)
  10    FORMAT(3X,'FIRST TYPE IN "I", THE PT. # YOU WISH TO
        1 REENTER')
        READ(5,*)I
        WRITE(5,20)
  20    FORMAT(3X,'NOW TYOE IN THE CORRESPONDING X,Y, AND Z
        1 ',/,'POSITIONS IN INCHES.')
        READ(5,*)ANS
        IF(ANS.EQ.'E')GOTO 100
        READ(5,*)IX,IY,IZ
  30    WRITE(5,40)
  40    FORMAT(3X,'NOW TYPE IN THE -Y- COUNTER READING
        1 FOR YOUR POSITION')
        READ(5,*)IYCNTS
        WRITE(1'1)I,IX,IY,IZ,IYCNTS
        CLOSE(UNIT=1)
  100   STOP
        END
```

SHIFT.FTN

```
C ***********************************************************

C ...  THE PROGRAM SHIFT REARRANGES THE STORED FORMAT OF THE
C ...  DATA IN THE FILE ARRAY.DAT CREATED BY RUNNING THE
C ...  PROGRAM COSPX.FTN, INSTVEL.FTN, VOLTS.FTN OR AVERAGE.
C....  FTN, FOR SUBSEQUENT USE IN THE PROGRAM VECTOR.FTN

C ...  THE COUNTER THAT APPEARS ON THE SCREEN AS SHIFT IS
C ...  RUN GOES FROM 1 TO 72 AND CAN TAKE CONSIDERABLE TIME
C ...  TO ACCOMPLISH THIS DEPENDING UPON THE NUMBER OF POINTS
C ...  BEING SHIFTED, (IE: AN X-PLANE HAS 11 X 6 OR 66 POINTS
C ...  WHEREAS AN ENTIRE CASE HAS 11 X 6 X 18 OR 1188 POINTS)

C ***********************************************************

        INTEGER*4 IDATA(18,6,11),ARRAY(114),ITEST(18,6,11)
        BYTE ID1(80),ID2(80)
10      WRITE(5,100)
100     FORMAT(' TYPE IN SIX LETTER SOURCE .DAT FILE NAME: ',$)
        READ(5,101,ERR=10,END=1000) I,(ID1(J),J=4,I+4)
 101    FORMAT(Q,12A1)

C ...  UNLESS THE UNIT NUMBER IN THE DATA TAKING PROGRAM
C ...  HAS BEEN CHANGED, THE UNIT NUMBER REQUESTED HERE
C ...  WILL BE "2".
        WRITE(5,102)
 102    FORMAT(' UNIT NUMBER WHICH SOURCE FILE IS IN:   ', $)
        READ(5,111) IU1
        WRITE(5,106)
        READ(5,107) NIVEL
 121    WRITE(5,104)
 104    FORMAT(' MIN/MAX #-POINTS ALONG X-Y-Z AXES (IE: 1,18,
        1 1,6,1,11) '$)
C ***  THE "IE" ABOVE IS FOR AN ENTIRE CASE
        READ(5,*,ERR=121,END=1000) IXS,IXE,IYS,IYE,IZS,IZE

C ...  THE ARRAY LENGTH ASKED FOR HERE WILL BE "72" UNLESS
C ...  THE DATA TAKING PROGRAM IS RECORDING MORE THAN 72
C ...  POINTS PER BLADE REVOLUTION OR UNLESS YOU ARE SHIFTING
C ...  AVERAGE DATA IN WHICH CASE THE ARRAY SIZE IS 1.
 106    FORMAT(' ARRAY LENGTH OF SINGLE DATA SET: ',  $)
 107    FORMAT(1I5)
 11     WRITE(5,108)
 108    FORMAT('TYPE IN SIX LETTER DESTINATION .DAT FILE NAME:
        1 ', $)
        READ(5,109,ERR=11,END=1000) I,(ID2(J),J=4,I+4)
 109    FORMAT(Q,12A1)
```

```
C ... THE UNIT NUMBER ASKED FOR HERE CAN BE ANY UNIT OTHER THAN
C ... THAT BEING USED BY THE SOURCE FILE AND NOT 5,6 OR 7.
        WRITE(5,110)
 110    FORMAT(' UNIT NUMER WHICH DESTINATION FILE IS TO BE
        1 STORED ',$)
        READ(5,111) IU2
 111    FORMAT(1I1)

        ID1(1)='S'
        ID2(1)='S'
        ID1(2)='Y'
        ID2(2)='Y'
        ID1(3)=':'
        ID2(3)=':'
        ID1(I+4)='.'
        ID2(I+4)='.'
        ID1(I+5)='D'
        ID2(I+5)='D'
        ID1(I+6)='A'
        ID2(I+6)='A'
        ID1(I+7)='T'
        ID2(I+7)='T'

        NPTS=0
C ... THESE THREE DO LOOPS JUST PERFORM THE MULTIPLICATION OF
C ... THE MAXIMUM X,Y,Z LIMITS TO GIVE THE NUMBER OF POINTS
C ... BEING SHIFTED.
        DO 12 I=IZS,IZE
        DO 13 K=IYS,IYE
        DO 14 J=IXS,IXE
        NPTS=NPTS+1
 14     CONTINUE
 13     CONTINUE
 12     CONTINUE
        WRITE(5,199) NPTS
 199    FORMAT(' #-POINTS=',I10,$)

        OPEN(UNIT=IU2,NAME=ID2,TYPE='NEW',ACCESS='DIRECT'
        1 ,FORM='UNFORMATTED',RECORDSIZE=NPTS)
        OPEN(UNIT=IU1,NAME=ID1,TYPE='OLD',ACCESS='DIRECT')
        DO 1 II=1,NIVEL
        III=0
        DO 2 K=IZS,IZE
        DO 3 I=IYS,IYE
        DO 4 J=IXS,IXE
        III=III+1
        READ(IU1'III) (ARRAY(JJ),JJ=1,NIVEL)
        IDATA(J,I,K)=ARRAY(II)
 4      CONTINUE
 3      CONTINUE
 2      CONTINUE
```

```
      WRITE(IU2'II) (((IDATA(K, I, J), K=IXS, IXE), I=IYS, IYE),
      1 J=IZS, IZE)
      WRITE(5, 300) ARRAY(II), IDATA(IXE, IYE, IZE), II
 300  FORMAT(1X, I10, 2X, I10, 3X, 'I = ', I2)
 1    CONTINUE

C ... THE FOLLOWING WRITE STATEMENT IS COMMENTED HERE
C ... BUT CAN BE USED BY REMOVING THE "C" TO PRINT OUT
C ... THE ENTIRE IDATA ARRAY WHICH CONTAINS THE 9-DIGIT
C ... CODED FORM OF THE X, Y, Z VELOCITIES.
C     WRITE(6, 301)IDATA
  301 FORMAT(3X, 6I10)
C
 1000 CONTINUE

      CLOSE(UNIT=IU1)
      CLOSE(UNIT=IU2)
      END
```

STORE.FTN

```
C *******************************************

C THE PROGRAM STORE.FTN IS A SIMPLE READ WRITE
C OPERATION WHICH SPLITS A WHOLE CASE OF DATA
C INTO TWO SEPARATE FILES IN ORDER TO ALLOW
C STORAGE ON TWO FLOPPY DISKS.  A CASE OF DATA
C USES 669 BLOCKS OF STORAGE AND A FLOPPY DISK
C HOLDS ONLY ABOUT 430.

C THE RESULT OF THE PROGRAM IS TWO DATA FILES
C CALLED HALF1.DAT AND HALF2.DAT WHICH CAN BE
C STORED SEPARATELY ON TWO FLOPPIES.  TO RE-
C CREATE THE FILE ARRAY.DAT FOR USE IN PLOTTING
C THE FOLLOWING PROCEDURE SHOULD BE FOLLOWED.

C TYPE IN "MERGE HALF2.DAT HALF1.DAT"
C THIS CREATES A FILE HALF1.DAT WHICH IS THE
C ADDITION OF HALF2 TO HALF1.
C NEXT JUST TYPE COPY HALF1.DAT ARRAY.DAT.

C ********************************************
        INTEGER*4 ARRAY(72),I
        OPEN(UNIT=2,NAME='ARRAY.DAT',TYPE='OLD',ACCESS='
        1 DIRECT')
        OPEN(UNIT=3,NAME='HALF2.DAT',TYPE='NEW',ACCESS='
        1 DIRECT',FORM='UNFORMATTED',RECORDSIZE=72)
        OPEN(UNIT=6,NAME='HALF1.DAT',TYPE='NEW',ACCESS='
        1 DIRECT',FORM='UNFORMATTED',RECORDSIZE=72)
        DO 100 I=1,763
        READ(2'I) (ARRAY(J),J=1,72)
        WRITE(6'I) (ARRAY(J), J=1,72)
100     CONTINUE
        DO 101 I=764,1188
        READ(2'I) (ARRAY(J),J=1,72)
        II=I-763
        WRITE(3'II) (ARRAY(J),J=1,72)
101     CONTINUE
        CLOSE(UNIT=2)
        CLOSE(UNIT=3)
        CLOSE(UNIT=6)
        STOP
        END
```

VECTOR.FTN

```
C *************************************************

C  THE PROGRAM VECTOR.FTN ALLOWS THE USER TO PLOT
C  THE INSTANTANEOUS VELOCITY VECTORS IN THE X, Y
C  OR Z PLANES.  IN ORDER TO USE THIS PROGRAM THE
C  DATA AS TAKEN IN THE TUNNEL MUST BE SHIFTED USING
C  THE PROGRAM SHIFT.FTN.  THIS REARRANGES THE DATA
C  IN A DIRECT ACCESS FILE WHICH HAS ALL THE VELOCITY
C  DATA FOR A SPECIFIC BLADE ANGLE INSTEAD OF ALL THE
C  ANGLES FOR ONE POINT.

C **************************************************

C  TO RUN VECTOR.FTN IT IS FIRST NECESSARY THAT THE
C  PROGRAM BE COMPILED AND LINKED.  SINCE THE DATA
C  IN THE PROGRAM IS DIRECT ACCESS THE ORDINARY "LINK"
C  COMMAND CANNOT BE USED.  THE FTB OR TKB DIRECT
C  ACCESS LINK PROCEDURES MUST BE USED.  THIS IS DONE
C  AS FOLLOWS:

C  YOU  FTB
C  COMPUTER  FTB>
C  YOU  VECTOR=VECTOR
C  COMPUTER  FTB>
C  YOU  /
C  COMPUTER  ENTER OPTIONS:
C  COMPUTER  FTB>
C  YOU  MAXBUF=4800
C  COMPUTER  FTB>
C  YOU  //

C ***** NOTE THAT THE BUFFER HAS BEEN DECLARED AS 4800
C ***** THIS IS THE SIZE NEEDED WHEN ACCESSING AN
C ***** ENTIRE CASE OF DATA.  THE BUFFER SIZE MUST BE
C ***** AT LEAST 4 TIMES THE RECORDSIZE OF THE FILE
C ***** 4 TIMES 1188 EQUALS 4752

C ***************************************************

      INTEGER*4 IDATA1(18,11),IDATA2(18,11)
      LOGICAL ITEST,IHRD
      COMMON /C1/IHRD,ITEST,IU,INVELS,INVELE,INVEL
      COMMON /C6/IXP,IYP,IHOLD
      IHOLD=1
      ITEST=.TRUE.
C *** THE SUBROUTINES ATTACH, INITT, TERM, CHRSIZ
C *** ,ANMARM, GRLOOK, NUMBUF, NUMDRW, AND KEYFUN
C *** ARE ALL PLOT 10 SYSTEM SUBROUTINES
      CALL ATTACH
      CALL INITT(960)
      CALL TERM(2,1024)
      CALL CHRSIZ(4)
```

```
C
        CALL ANMARM
        CALL GRLOOK(1)
        CALL NUMBUF(2)
        CALL NUMDRW(3)
        CALL KEYFUN(1)
C
        CALL ANMODE
        CALL ASK1(IFILE)
        IF(IFILE.EQ.2)WRITE(5,15)
C......SHIFT.FTN IS A PROGRAM WHICH REARRANGES THE DATA
C......TO AN X,Y,Z AT A PARTICULAR AZIMUTH FOR EACH POINT
C......RATHER THAN THE AS TAKEN FORM OF X,Y,Z FOR ALL
C......AZIMUTH POSITIONS AT EACH POINT.
C
15      FORMAT(' TYPE IN "RUN SHIFT"')
        IF(IFILE.EQ.2) GO TO 1000
20      CONTINUE
        CALL FILE
        CALL ASK2
        CALL NEWPAG
        CALL BORDR
        CALL TUNNEL
        IHOLD=2
        CALL CZAXIS(0)
        CALL HOME
        CALL ANMODE
        CALL WHERE
        CALL SETPAR
        CALL INFOIN
        CALL STPANM
        IBUF=1
        CALL NEWPAG
C
        JJ=INVELS
        GO TO 91
19      IF(IHRD) JJ=JJ+INVEL
        IF((JJ.GT.INVELE).OR.(JJ.EQ.INVELE)) JJ=INVELS
91      IF(IHRD) CALL NEWPAG
        CALL CZAXIS(0)
        CALL INFOUT
        CALL CZAXIS(1)
        CALL ORIGIN(JJ)
        IF(IHRD) GO TO 2002
C *** INVELS=STARTING BLADE ANGLE
C *** INVELE=ENDING BLADE ANGLE
C *** INVEL=BLADE ANGLE INCREMENT
```

```
      DO 2000 JJ=INVELS, INVELE, INVEL
C
2002  CONTINUE
      CALL DATA(JJ, I, J, IDATA1, IDATA2)
      IF(.NOT.IHRD)CALL CZAXIS(2)
      IF(IHRD) CALL CZAXIS(0)
      IF(.NOT.IHRD) CALL FRAME(IBUF, 1)
      CALL PLOT2(I, J, IDATA1, IDATA2)
      IF(IHRD) GO TO 2000
      CALL ANIMAT(IBUF)
      IBUF=IBUF+1
      IF(IBUF.EQ.3) IBUF=1
2000  CONTINUE
C
      IF(IHRD) CALL STPANM
      READ(5,1928) IQAZ
1928  FORMAT(1A1)
      CALL STPANM
C
      CALL HOME
      CALL ANMODE
      IF(IHRD) WRITE(5,8991)
8991  FORMAT(' NEXT PLOT? Y-N ',$)
      READ(5,2001) IUYT
      IF(IUYT.EQ.'Y') GO TO 19
2001  FORMAT(1A1)
      WRITE(5,9000)
      READ(5,9001) IDUM
9000  FORMAT(' ANOTHER RUN?: Y-N ',$)
9001  FORMAT(1A1)
      IF(IDUM.EQ.'N') GO TO 1001
      WRITE(5,21)
21    FORMAT(' FROM THE SAME SOURCE FILE?: Y-N ',$)
      READ(5,22) ITES
 22   FORMAT(1A1)
      IF(ITES.EQ.'Y') ITEST=.FALSE.
      IF(ITES.EQ.'N') ITEST=.TRUE.
      IF(ITES.EQ.'N') CLOSE(UNIT=IU)
      GO TO 20
 1001 CLOSE(UNIT=IU)
      CALL STPANM
 1000 CONTINUE
      END
C *************************************
```

```
C THE SUBROUTINE FILE ASKS THE USER THE
C FILE NAME OF THE DATA SET CREATED BY
C THE PROGRAM SHIFT.FTN.
        SUBROUTINE FILE
        BYTE ID(20)
        LOGICAL ITEST
        COMMON /C1/IHRD, ITEST, IU, INVELS, INVELE, INVEL
        IF(.NOT.ITEST) GO TO 1001
 17     WRITE(5,19)
 19     FORMAT(' NAME OF SIX LETTER .DAT DIRECT ACCESS SOURCE
        1  FILE:  ',$)
        READ(5,18,ERR=17,END=1001) I,(ID(J),J=4,I+4)
 18     FORMAT(Q,12A1)
        WRITE(5,16)
 16     FORMAT(' UNIT NUMBER SOURCE .DAT FILE IS STORED IN:  ',$)
        READ(5,*) IU
        ID(1)='S'
        ID(2)='Y'
        ID(3)=':'
        ID(I+4)='.'
        ID(I+5)='D'
        ID(I+6)='A'
        ID(I+7)='T'
        OPEN(UNIT=IU,NAME=ID,TYPE='OLD',ACCESS='DIRECT')
C
 1001   RETURN
        END
C ****************************************
        SUBROUTINE INFOIN
        BYTE ID1(200)
        INTEGER IDPL(10)
        COMMON /C2/ID1,IDPL
        WRITE(5,1)
        WRITE(5,2)
        WRITE(5,3)
 1      FORMAT(' TYPE IN ALL INFORMATION ABOUT PLOT')
 2      FORMAT(' TO BE DISPLAYED WITH PLOT ')
 3      FORMAT(' LINE BY LINE, AS MANY LINES AS NEEDED.')
C
        III=1
        ISS=1
        WRITE(5,5)
 5      FORMAT('TYPE IN INFO.  WHEN "->" APPEARS')
        WRITE(5,6)
 6      FORMAT('TYPE IN:  **  TO TERMINATE INPUT.')
 900    WRITE(5,7)
        III=III+1
 7      FORMAT(' ->', $)
        READ(5,8,END=1001) I,(ID1(II),II=ISS,(ISS+I-1))
 8      FORMAT(Q,35A1)
```

```
      IF(ID1(ISS).EQ. '*' .AND. ID1(ISS+1).EQ.'*') GO TO 901
      ISS=ISS+1
      IDPL(III)=I
      GO TO 900
 901  IDPL(1)=III-1
 1001 RETURN
      END
C ***************************************

C *** IGD=THE DISTANCE BETWEEN GRID POINTS
C *** ISCX=THE SCREEN X MAGNITUDE FACTOR FOR VECTORS
C *** ISCY=THE SCREEN Y MAGNITUDE FACTOR FOR VECTORS
C *** ISKX=THE SKEW FACTOR IN SCREEN X-DIRECT.
C *** ISKY=THE SKEW FACTOR IN SCREEN Y-DIRECT.
C *** IX=X COORD. FROM SCREEN LOWER LEFT
C *** IY=Y COORD. FROM SCREEN LOWER LEFT

C ********************************************
      SUBROUTINE SETPAR
      COMMON /C3/IXA,IYB,IL,IV,IP
      COMMON /C5/IGD,ISCX,ISCY,ISKX,ISKY,IX,IY
      COMMON /C6/IXP,IYP,IHOLD
      COMMON /C1/IHRD,ITEST,IU,INVELS,INVELE,INVEL
      WRITE(5,15)
      WRITE(5,16)
 20   WRITE(5,12)
      WRITE(5,13)
      WRITE(5,14)
      IX=IX-100
      WRITE(5,8)
      WRITE(5,7)
      WRITE(5,9)
      WRITE(5,15)
 12   FORMAT(' IGRID=SIZE OF GRID (TRY 55)')
 15   FORMAT(' ')
 16   FORMAT('********** SET PARAMETERS ***********')
 9    FORMAT(' IX,IY= RASTER REFERENCE POINTS: (IE:100,100)')
 13   FORMAT(' SCALEX=SCALING FACTOR ON X-MAGNITUDE')
 14   FORMAT(' SCALEY=SCALING FACTOR ON Y-MAGNITUDE')
 8    FORMAT(' SKEWX=AMOUNT OF SKEW SUBTRACTION ON "X" ')
 7    FORMAT(' SKEWY=AMOUNT OF SKEW SUBTRACTION ON "Y" ')
      WRITE(5,10)
      WRITE(5,11) IGD,ISCX,ISCY,ISKX,ISKY,IX,IY
 11   FORMAT(1X,7I7)
 10   FORMAT(' INPUT:IGRID,SCALEX,SCALEY,SKEWX,SKEWY,IX,IY')
      READ(5,*,END=1000,ERR=20)IGD,ISCX,ISCY,ISKX,ISKY,IX,IY
      IX=IX+100
 1000 CONTINUE
      WRITE(5,15)
```

```
C *** NOTE: IF PLOTTING OF AVERAGE DATA IS REQUIRED, THE VALUES
C ***    IN THE FOLLOWING STATEMENT WILL BE 1,1,1 WHEN THEY ARE
C ***    ASKED FOR IN THE INSTRUCTIONS.

        WRITE(5,30) INVELS,INVELE,INVEL
 30     FORMAT(' LENGTH OF SINGLE DATA POINT ARRAY(1,72,1): ',
        1 3I3,1X,$)
        READ(5,*,END=1001,ERR=1000) INVELS,INVELE,INVEL
 1001   RETURN
        END

C ****************************************

        SUBROUTINE INFOUT
        BYTE ID1(200)
        INTEGER IDPL(10)
        COMMON /C2/ID1,IDPL
        CALL CZAXIS(0)
        ISS=1
        DO 910 K=2,IDPL(1)
        CALL MOVABS(770,(620-K*10))
        CALL ANMODE
        WRITE(5,850) (ID1(I),I=ISS,(IDPL(K)+ISS-1))
 850    FORMAT(Q,35A1)
        ISS=ISS+IDPL(K)
 910    CONTINUE
        RETURN
        END

C ****************************************

        SUBROUTINE MOVE3(IX,IY,IZ)
        REAL OUTX,OUTY,OUTZ,TEMP,RMAT(4,4)
        INTEGER IX,IY,IZ,IOUTX,IOUTY
        COMMON /C6/IXP,IYP,IH
        COMMON /C8/IXHP,IYHP,I
        COMMON /RMAT/RMAT
        IF(IH.EQ.2) GO TO 1234
        IF(IH.EQ.1)IXP=IXHP
        IF(IH.EQ.1)IYP=IYHP
 1234   CONTINUE
        X=FLOAT(IX)
        Y=FLOAT(IY)
        Z=FLOAT(IZ)
        OUTX=X*RMAT(1,1)+Y*RMAT(2,1)+Z*RMAT(3,1)+RMAT(4,1)
C
        OUTY=X*RMAT(1,2)+Y*RMAT(2,2)+Z*RMAT(3,2)+RMAT(4,2)
```

```
C
      OUTZ=X*RMAT(1,3)+Y*RMAT(2,3)+Z*RMAT(3,3)+RMAT(4,3)
C
      TEMP=OUTX*RMAT(1,4)+OUTY*RMAT(2,4)+OUTZ*RMAT(3,4)+
     1    RMAT(4,4)
      OUTX=OUTX/TEMP
      OUTY=OUTY/TEMP
      IOUTX=IFIX(OUTX)+IXP
      IOUTY=IFIX(OUTY)+IYP
      CALL MOVABS(IOUTX,IOUTY)
      RETURN
      END

C ***************************************

      SUBROUTINE DRAW3(IX,IY,IZ)
      REAL OUTX,OUTY,OUTZ,TEMP,RMAT(4,4)
      INTEGER IX,IY,IZ,IOUTX,IOUTY
      COMMON /C6/IXP,IYP,IH
      COMMON /C8/IXHP,IYHP,I
      COMMON /RMAT/RMAT
      IF(IH.EQ.2) GO TO 1234
      IF(IH.EQ.1)IXP=IXHP
      IF(IH.EQ.1)IYP=IYHP
 1234 CONTINUE
      X=FLOAT(IX)
      Y=FLOAT(IY)
      Z=FLOAT(IZ)
      OUTX=X*RMAT(1,1)+Y*RMAT(2,1)+Z*RMAT(3,1)+RMAT(4,1)
C
      OUTY=X*RMAT(1,2)+Y*RMAT(2,2)+Z*RMAT(3,2)+RMAT(4,2)
C
      OUTZ=X*RMAT(1,3)+Y*RMAT(2,3)+Z*RMAT(3,3)+RMAT(4,3)
C
      TEMP=OUTX*RMAT(1,4)+OUTY*RMAT(2,4)+OUTZ*RMAT(3,4)+
     1    RMAT(4,4)
      OUTX=OUTX/TEMP
      OUTY=OUTY/TEMP
      IOUTX=IFIX(OUTX)+IXP
      IOUTY=IFIX(OUTY)+IYP
      CALL DRWABS(IOUTX,IOUTY)
      RETURN
      END
```

```
C ***************************************
      SUBROUTINE INIMAT(PX, PY, PZ, PHE, RX, RY, RZ)
      REAL ALPHA, BETA, GAMMA, RMAT(4, 4), PS, PC, N1, N2, N3
      INTEGER PX, PY, PZ, PHE, RX, RY, RZ
      INTEGER*4 ITEMP
      COMMON /C6/IXP, IYP, IH
      COMMON /C8/IXHP, IYHP, I
      COMMON /RMAT/RMAT
      IF(IH. EQ. 2) GO TO 1234
      IF(IH. EQ. 1) IXP=IXHP
      IF(IH. EQ. 1) IYP=IYHP
 1234 CONTINUE

      ITEMP=PX**2+PY**2+PZ**2
      TEMP=FLOAT(ITEMP)
      TEMP=SQRT(TEMP)
      N1=FLOAT(PX)/TEMP
      N2=FLOAT(PY)/TEMP
      N3=FLOAT(PZ)/TEMP
      PS=SIN(FLOAT(PHE)/180. 0*3. 14159)
      PC=COS(FLOAT(PHE)/180. 0*3. 14159)
      RMAT(1, 1)=N1*N1+(1. -N1*N1)*PC
      RMAT(1, 2)=N1*N2*(1. -PC)+N3*PS
      RMAT(1, 3)=N1*N3*(1. -PC)-N2*PS
      RMAT(1, 4)=0. 0
      IF(RX. NE. 0) RMAT(1, 4)=1. /FLOAT(RX)
      RMAT(2, 1)=N1*N2*(1. -PC)-N3*PS
      RMAT(2, 2)=N2*N2+(1. -N2*N2)*PC
      RMAT(2, 3)=N2*N3*(1. -PC)+N1*PS
      RMAT(2, 4)=0. 0
      IF(RY. NE. 0) RMAT(2, 4)=1. /FLOAT(RY)
      RMAT(3, 1)=N1*N3*(1. -PC)+N2*PS
      RMAT(3, 2)=N2*N3*(1. -PC)-N1*PS
      RMAT(3, 3)=N3*N3+(1. -N3*N3)*PC
      RMAT(3, 4)=0. 0
      IF(RZ. NE. 0) RMAT(3, 4)=1. 0/FLOAT(RZ)
      RMAT(4, 1)=0. 0
      RMAT(4, 2)=0. 0
      RMAT(4, 3)=0. 0
      RMAT(4, 4)=1. 0
      RETURN
      END
```

```
C *****************************************

C
        SUBROUTINE BORDR
        CALL HOME
        CALL DRWABS(0,0)
        CALL DRWABS(1023,0)
        CALL DRWABS(1023,780)
        CALL DRWABS(0,780)
        RETURN
        END
C *****************************************

        SUBROUTINE ASK1 (K)
        COMMON /C1/IHRD,ITEST,IU,INVELS,INVELE,INVEL
        WRITE(5,30)
 30     FORMAT(' HAS THE DAT BEEN SHIFTED? "Y" OR "N"' $)
        READ(5,31) SH
 31     FORMAT(1A1)
        IF(SH.EQ.'Y')K=1
        IF(SH.EQ.'N')K=2
        RETURN
        END

C *****************************************

        SUBROUTINE ASK2
        COMMON /C1/IHRD,ITEST,IU,INVELS,INVELE,INVEL
        WRITE(5,33)
 33     FORMAT(' HARDCOPIES OR ANIMATION?: "H" OR "A"',$)
        READ(5,34) IWAY
 34     FORMAT(1A1)
        IF(IWAY.EQ.'H') IHRD=.TRUE.
        IF(IWAY.EQ.'A') IHRD=.FALSE.
        RETURN
        END

C *****************************************

C  THE INSTRUCTIONS FOR THIS PART ARE FAIRLY
C  CLEAR EXCEPT FOR ONE POINT.  IF YOU ARE
C  ACCESSING A WHOLE CASE SPECIFY THE BEGINNING
C  AND ENDING LOCATIONS OF THE WHOLE CASE EVEN
C  IF YOU ARE ONLY PLOTTING ONE PLANE.
```

```
C *** IXS=X STARTING LOCATION
C *** IXE=X ENDING LOCATION
C *** IYS=Y STARTING LOCATION
C *** IYE=Y ENDING LOCATION
C *** IZS=Z STARTING LOCATION
C *** IZE=Z ENDING LOCATION
C *** IP=PLANE X,Y OR Z (1,2 OR 3)
C *** IL=LOCATION OF PLANE
C ******************************************

        SUBROUTINE WHERE
        COMMON /C3/IXA, IYB, IL, IV, IP
        COMMON /C4/IXS, IXE, IYS, IYE, IZS, IZE
 25     WRITE(5, 24)
        WRITE(5, 33) IXS, IXE, IYS, IYE, IZS, IZE
 24     FORMAT(' MIN/MAX POINTS ALONG XYZ AXES TO BE VIEWED
        1      (IE: 9, 9, 1, 11, 1, 6)')
 33     FORMAT(1X, 6I4)
        WRITE(5, 34)
 34     FORMAT(1X, $)
        READ(5, *, ERR=25, END=69) IXS, IXE, IYS, IYE, IZS, IZE
 69     WRITE(5, 70) IP
 70     FORMAT(' SPECIFY PLANE "X" OR "Y" OR "Z" ', A1, $)
        READ(5, 71, END=1000) IP
 71     FORMAT(1A1)
        IF(IP.EQ.0) GO TO 1001
 1000   IF(IP.EQ. 'X') GO TO 200
        IF(IP.EQ. 'Y') GO TO 400
        IF(IP.EQ. 'Z') GO TO 300
        WRITE(5, 3332)
 3332   FORMAT(' SPECIFY "X" "Y" OR "Z" ', $)
        READ(5, 71, END=1001) IP
 200    CONTINUE
 201    WRITE(5, 72) IP, IL
 72     FORMAT(' LOCATION ALONG ', A1, '-AXIS  ', I2,   $)
        READ(5, 199, END=74) IL
 199    FORMAT(1I4)
        IF((IL.LT.1).OR.(IL.GT.18))GO TO 201
        IYB=IZE
        IXA=IYE
        GO TO 74
 300    WRITE(5, 72) IP, IL
        READ(5, 199, END=74) IL
        IF((IL.GT.6).OR.(IL.LT.1)) GO TO 300
        IXA=IXE
        IYB=IYE
        GO TO 74
 400    WRITE(5,72) IP,IL
        READ(5,199,END=74) IL
        IF((IL.LT.1).OR.(IL.GT.11)) GO TO 400
        IXA=IXE
        IYB=IZE
 74     WRITE(5,26) IV
 26     FORMAT(' VECTORS XY,XZ, OR YZ:
        1  TYPE 1,2 OR 3 RESPECTIVELY   ',I1,1X, $)
        READ(5,*,END=1001) IV
 1001   RETURN
        END

C *****************************************

C THIS SUBROUTINE DOES THE ACTUAL DRAWING OF THE
C VECTORS.

C ******************************************

        SUBROUTINE PLOT2(IIX,IIY,IDATA1,IDATA2)
        INTEGER*4 IMX,IMY,IDATA1(18,11),IDATA2(18,11)
        COMMON /C5/IGD,ISCX,ISCY,ISKX,ISKY,IX,IY
        DO 1 I=1,IIY
        IYY=IY+IGD*I
        DO 2 J=1,IIX
        IXX=IX+IGD*J
        IMX=IDATA1(J,I)/ISCX+ISKX
        IMY=IDATA2(J,I)/ISCY+ISKY
        CALL MOVABS(IXX,IYY)
        CALL DRWREL(IMX,IMY)
 2      CONTINUE
 1      CONTINUE
        RETURN
        END

C ****************************************

C.....THIS SUBROUTINE ASSIGNS THE APPROPRIATE VALUES
C.....FROM THE THREE-DIMENSIONAL ARRAY, IDATA TO THE
C.....TWO-DIMENSIONAL ARRAYS, IDATA1 AND IDATA2 FOR
C.....SUBSEQUENT PLOTTING USING SUBROUTINE PLOT2.

C ************************************************

        SUBROUTINE DATA(JJ,I,J,IDATA1,IDATA2)
        INTEGER*4 IDATA(18,6,11),IZ,IZ1,IZ2,IZH
        INTEGER*4 IDATA1(18,11),IDATA2(18,11)
        COMMON /C3/IXA,IYB,IL,IV,IP
        COMMON /C1/IHRD,ITEST,IU,INVELS,INVELE,INVEL
        COMMON /C4/IXS,IXE,IYS,IYE,IZS,IZE
        READ(IU'JJ) (((IDATA(I,J,K),I=IXS,IXE),J=IZS,IZE),K=IYS,IYE)
  749   IF(IP.EQ.'X') GO TO 700
        IF(IP.EQ.'Y') GO TO 701
```

```
C
C.....THE FOLLOWING DO LOOPS ARE FOR THE -- Z -- PLANE
C
      DO 750 J=1,IYB
      DO 751 I=1,IXA
      IZ=IDATA(I,IL,J)
      CALL DECODE(IV,IZ,IZ1,IZ2)
C     IF(J.EQ.1) IZH=IZ2
      IDATA1(I,J)=IZ1-499
      IDATA1(I,J)=-IDATA1(I,J)
      IDATA2(I,J)=IZ2-499
      IDATA2(I,J)=-IDATA2(I,J)
 751  CONTINUE
 750  CONTINUE
      I=I-1
      J=J-1
      RETURN
C
C.....THE FOLLOWING DO LOOPS ARE FOR THE -- X -- PLANE
C
 700  DO 760 J=1,IXA
      DO 761 I=1,IYB
      IZ=IDATA(IL,I,J)
      CALL DECODE(IV,IZ,IZ2,IZ1)
C     IF(J.EQ.1) IZH=IZ2
      IDATA1(I,J)=IZ1-499
      IDATA1(I,J)=-IDATA1(I,J)
      IDATA2(I,J)=IZ2-499
      IDATA2(I,J)=-IDATA2(I,J)
 761  CONTINUE
 760  CONTINUE
      I=I-1
      J=J-1
      RETURN
C
C.....THE FOLLOWING DO LOOPS ARE FOR THE -- Y -- PLANE
C
 701  DO 770 J=1,IYB
      DO 771 I=1,IXA
      IZ=IDATA(I,J,IL)
      CALL DECODE(IV,IZ,IZ1,IZ2)
C     IF(J.EQ.1)IZH=IZ2
      IDATA1(I,J)=IZ1-499
      IDATA1(I,J)=-IDATA1(I,J)
      IDATA2(I,J)=IZ2-499
      IDATA2(I,J)=-IDATA2(I,J)
 771  CONTINUE
 770  CONTINUE
      I=I-1
      J=J-1
      RETURN
      END
```

```
C ****************************************
C  THIS SUBROUTINE DECODES THE DATA FROM ITS
C  STORED 9-DIGIT INTEGER FORM.
C *****************************************


        SUBROUTINE DECODE(IV, IZ, IZ1, IZ2)
        INTEGER*4 IZ, IZ1, IZ2, IZ3, IZHH
        IZ1=IZ/1E6
        IZ2=(IZ-IZ1*1E6)/1E3
        IF(IV. EQ. 1)RETURN
        IZHH=IZ2
        IZ3=IZ-IZ1*1E6-IZ2*1E3
        IZ2=IZ3
        IF(IV. EQ. 2) RETURN
        IZ1=IZHH
        RETURN
        END
C
C ****************************************

C THIS SUBROUTINE DRAWS THE SCHEMATIC OF THE
C TUNNEL.

C *****************************************

        SUBROUTINE TUNNEL
        COMMON /C8/IXHP, IYHP, IHOL
        IXHP=680
        IYHP=490
        IHOL=1
        CALL INIMAT(1, 1, 1, 50, 0, 0, 0)
        MAG=24
        DO 10 II=0, 90, 15
        IX=IFIX(MAG*COS(FLOAT(II)*3. 14159/180. ))
        IY=IFIX(MAG*SIN(FLOAT(II)*3. 14159/180. ))
        CALL MOVE3(IX, IY, 0)
        CALL DRAW3(-IX, -IY, 0)
        CALL MOVE3(IX, -IY, 0)
        CALL DRAW3(-IX, IY, 0)
 10     CONTINUE
        IX=200
        IY=140
        IZ=120
        NX=-180
        NZ=-60
```

```
C
      CALL MOVE3(+IX,+IY,+IZ)
      CALL DRAW3(+IX,-IY,+IZ)
      CALL DRAW3(NX,-IY,+IZ)
      CALL DRAW3(NX,+IY,+IZ)
      CALL DRAW3(+IX,+IY,+IZ)
C
      CALL DRAW3(+IX,+IY,NZ)
      CALL DRAW3(+IX,-IY,NZ)
      CALL DRAW3(+IX,-IY,+IZ)
C
      CALL MOVE3(+IX,-IY,NZ)
      CALL DRAW3(NX,-IY,NZ)
      CALL DRAW3(NX,+IY,NZ)
      CALL DRAW3(+IX,+IY,NZ)
C
      CALL MOVE3(NX,+IY,NZ)
      CALL DRAW3(NX,+IY,+IZ)
      CALL MOVE3(NX,-IY,+IZ)
      CALL DRAW3(NX,-IY,NZ)
C
      IX=-160
      DO 1 I=1,18
      CALL MOVE3(IX,-IY,IZ)
      CALL DRAW3(IX,-IY,(IZ-10))
      CALL DRAW3(IX,-IY,(IZ+10))
      CALL MOVE3(IX,-IY,(IZ+20))
      CALL ANMODE
      WRITE(5,20)I
 20   FORMAT(1X,I2)
 1    IX=IX+20
      CALL MOVE3(0,-IY,(IZ+43))
      CALL ANMODE
      WRITE(5,21)
 21   FORMAT(' "X" ')
C
      IX=200
      IY=-100
      DO 2 I=1,11
      CALL MOVE3(IX,IY,IZ)
      CALL DRAW3((IX-10),IY,IZ)
      CALL DRAW3((IX+10),IY,IZ)
      CALL MOVE3((IX+20),(IY+8),IZ)
      CALL ANMODE
      WRITE(5,20)I
 2    IY=IY+20
      CALL MOVE3((IX+35),0,IZ)
      CALL ANMODE
      WRITE(5,22)
 22   FORMAT(' "Y" ')
C
```

```
          IY=-140
          IX=-180
          IZ=80
          DO 5 I=1,6,1
          CALL MOVE3(IX,IY,IZ)
          CALL DRAW3((IX+10),IY,IZ)
          CALL DRAW3((IX-10),IY,IZ)
          CALL MOVE3((IX+20),IY,(IZ-30))
          CALL ANMODE
          WRITE(5,20) I
 5        IZ=IZ-20
          CALL MOVE3((IX-35),IY,0)
          CALL ANMODE
          WRITE(5,23)
 23       FORMAT(' "Z" ')
          RETURN
          END
C
C ********************************************
          SUBROUTINE ORIGIN(JJ)
          COMMON /C5/IGD,ISX,ISY,ISKX,ISKY,IX,IY
          COMMON /C1/IHRD,ITEST,IU,INVEL1,INVEL2,INVEL3
          COMMON /C3/IXA,IYB,IL,IV,IP
          IDEG=JJ*5
          IF(IP.EQ.'Y',GO TO 10
          IF(IP.EQ.'Z')GO TO 10
          DO 1 I=1,11
          IYY=IY+IGD*I
          IYY=IYY-4
          DO 2 J=1,6
          IXX=IX+IGD*J
          IXX=IXX-2
          CALL MOVABS(IXX,IYY)
          CALL ANCHO(43)
 2        CONTINUE
 1        CONTINUE
          GO TO 20
  10      DO 11 I=1,IYB
          IYY=IY+IGD*I
          IYY=IYY-4
          DO 12 J=1,IXA
          IXX=IX+IGD*J
          IXX=IXX-2
          CALL MOVABS(IXX,IYY)
          CALL ANCHO(43)
  12      CONTINUE
  11      CONTINUE
 20     CONTINUE
        IF(.NOT.IHRD)RETURN
        CALL MOVABS(770,200)
        CALL CZAXIS(0)
        CALL ANMODE
        WRITE(5,3)IP,IL
3       FORMAT(' PLOT OF ',A1,'-PLANE AT LOCATION ',I2)
        CALL MOVABS(770,190)
        CALL ANMODE
        WRITE(5,4)IDEG
4       FORMAT(' AT BLADE ANGLE OF ',I3,' DEGREES')
        RETURN
        END
```

VOLTS.FTN

```
C ******************************************

C THE PROGRAM VOLTS.FTN IS A DATA AQUISITION
C PROGRAM WHICH CREATES A FILE FOR006.DAT
C CONTAINING THE VOLTAGES FOR THE AVERAGE OF
C THE 14 BLADE REVOLUTIONS.
C THESE ARE AVA, AVB, AND AVC IN THE PROGRAM.

C THE PROGRAM OPERATES ON THE DATA STORED IN
C THE DIRECT ACCESS FILE BUFFER.DAT.  BUFFER
C .DAT IS CREATED USING THE PROGRAM MAKEBUF.
C FTN.  THUS ONCE BUFFER HAS BEEN CREATED
C THERE IS NO NEED TO USE THE A/D CONVERTER
C FOR THIS PROGRAM.

C ONCE YOU HAVE FOR006.DAT USE EASY GRAPHING
C TO DO THE PLOTTING.  YOU MUST FIRST COPY
C FOR006.DAT TO PT.DAT, THEN TYPE EZG AND
C THEN RUN VEL.  OF COURSE YOU WILL NEED TO
C CHANGE THE SCALES AND LABELS.

C *******************************************

        INTEGER*4 ARRAY(72),NLOC,NVEL,IVX,IVY,IVZ
        1 ,I,IX,IY,IZ
        INTEGER BUFFER(1024,3)

C *** THE ARRAYS EAANG,EBANG, AND ECANG ARE STORING THE
C *** VELOCITIES FOR THE 14 REVS.

        DIMENSION EAANG(14),EBANG(14),ECANG(14),V(3),U(3)
        1 ,ASIN(3),AA(72),SUM(14),BB(72),CC(72)
        I=1
        IND=1
        OPEN(UNIT=1,NAME='BUFFER.DAT',TYPE='OLD',
        1 ACCESS='DIRECT')
        READ(1'1) (BUFFER(N,1),N=1,1024)
        READ(1'2) (BUFFER(N,2),N=1,1024)
        READ(1'3) (BUFFER(N,3),N=1,1024)
        WRITE(5,5000)
5000    FORMAT(3X,' BUFFER WAS FILLED')
```

```
        DO 85 J=1,72
C          (72=360/5)
        DO 9 M=1,3
        IA=J
        DO 8 JJ=1,14
C          14-REVOLUTIONS
        IF(M.EQ.1) EAANG(JJ)=FLOAT(BUFFER(IA,1))/(32768./8.)
        IF(M.EQ.2) EBANG(JJ)=FLOAT(BUFFER(IA,2))/(32768./8.)
        IF(M.EQ.3) ECANG(JJ)=FLOAT(BUFFER(IA,3))/(32768./8.)
        IA=IA+72
8       CONTINUE
        IF(M.EQ.1) CALL AVERAG(EAANG,AVA)
        IF(M.EQ.2) CALL AVERAG(EBANG,AVB)
        IF(M.EQ.3) CALL AVERAG(ECANG,AVC)
        IF(M.EQ.1) AA(J)=AVA
        IF(M.EQ.2) BB(J)=AVB
        IF(M.EQ.3) CC(J)=AVC
        WRITE(5,*)AVA,AA(J)
9       CONTINUE
C
85      CONTINUE
C
990     I=I+1
        WRITE(5,489)
489     FORMAT(4X,'DO YOU WANT AVA, AVB, OR AVC?')
        READ(5,567)ANSW
567     FORMAT(1A3)
        IF(ANSW.EQ.'AVB') WRITE(6,*)BB
        IF(ANSW.EQ.'AVC') WRITE(6,*)CC
        WRITE(6,*)AA
        WRITE(5,*)AA
        DO 1001 IU=1,3
1001    CLOSE(UNIT=IU)
        CLOSE(UNIT=6)
        STOP
        END

        SUBROUTINE AVERAG(A,AV)
        DIMENSION A(14),SUM(14)
        SUM(1)=A(1)
        DO 9 I=1,13
 9      SUM(I+1)=SUM(I)+A(I+1)
        AV=SUM(14)/14.
        RETURN
        END
```

APPENDIX D

FILMED ANIMATION OF VECTOR PLOTS

In order to more graphically illustrate the instantaneous nature of the data of Figures 4-4 through 4-6 it was decided to film the vector plots as they appear on the screen of a TEKTRONIX 4014 graphics computer terminal. At the time of this writing seven planes of data have been filmed; the X-Y planes at Z/R = 0.2 and 0.4; the X-Z planes at Y/R = 0.0, -0.8 and -0.53; and the Y-Z planes at X/R = -0.8 and -1.07. Each plane shows the flow every 5° of rotor rotation from 0° to 355°.

The dim light condition made it necessary to film each "instantaneous" plot in the single frame exposure mode. A 16 mm camera was used with a shutter speed of 1/4 second per frame focused at a distance of 2 feet from the screen. Twenty-five frames were exposed for each "instantaneous plot" which when projected at 24 frames per second gives a viewing time of about 1 second per plot. Thus one rotor revolution is projected in about 75 seconds.